AD-A048 554 ARMY CONCEPTS ANALYSIS AGENCY BETHESDA MD F/G 15/7

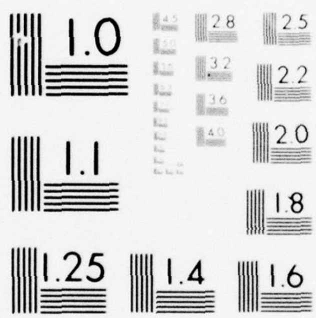
COMPUTER ASSISTED MATCH PROGRAM (CAMP),(U)

AUG 76 G L MARTIN, E R MONTAGNE

UNCLASSIFIED CAA-D-76-5 NL

1 OF 4

AD A048554


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

AD-A048554

COMPUTER ASSISTED SEARCH PROGRAM

(CASP)

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

(1)

DOCUMENTATION

CAA-D-76-5

AD-A

COMPUTER ASSISTED MATCH PROGRAM

(CAMP)

AUGUST 1976

PREPARED BY

JOINT AND STRATEGIC FORCES DIRECTORATE

US ARMY CONCEPTS ANALYSIS AGENCY

8120 Woodmont Avenue

Bethesda, Maryland 20014

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CAA-D-76-5	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Computer Assisted Match Program (CAMP)		5. TYPE OF REPORT & PERIOD COVERED Model Documentation
		6. PERFORMING ORG. REPORT NUMBER CAA-D-76-5
7. AUTHOR(s) Mr Gary L Martin MAJ Ernest R Montagne Jr		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Concepts Analysis Agency Joint and Strategic Forces Directorate 8120 Woodmont Avenue, Bethesda, MD 20014		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Concepts Analysis Agency Joint and Strategic Forces Directorate 8120 Woodmont Avenue, Bethesda, MD 20014		12. REPORT DATE 13 August 1976
		13. NUMBER OF PAGES 314
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) NA		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE NA

16. DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Force structure, Force match, Movement requirements, Strategic mobility

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

CAMP is a data processing system designed to be used as a tool in force structuring and analysis and strategic mobility analysis. CAMP contains two major functions: Force match algorithm (FMA) and Movement Requirements Generator. CAMP interfaces DA force planning files (Force Accounting System), the FASTALS theater roundout model, and various logistical data files to produce movement requirements in the Mobility Requirements for Staff Analysis system (MORSA) and/or SMOBSMOD format.

UNCLASSIFIED

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered) UNCLASSIFIED

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

CHAPTER I
SYSTEM INTRODUCTION

1. Purpose. The purpose of this documentation is to describe the Computer Assisted Match program (CAMP) Model in sufficient detail to allow use of CAMP by interested personnel.

2. Background

a. Various strategic mobility models require the input of force movement requirements in varying levels of detail.

b. Several studies conducted at the Concepts Analysis Agency (CAA) require production of input for the Strategic Mobility Simulation Model (SMOBSMOD), used at CAA, or the Mobility Requirements for Staff Analysis (MORSA) system, used at Joint Chiefs of Staff level. Automated production of such input is dictated by the volume of data required (typically, to 20,000 cards for MORSA). The CAMP was developed to meet this requirement.

3. General System Description. The CAMP is a data processing system developed at CAA. The system consists of two major functions as shown in Figure I-1: Force Match Algorithm (FMA) and Movement Requirements Generator (MRG). Each function includes a series of programs, primarily written in FORTRAN, but with some COBOL programs as well. The objectives of CAMP are to compare an actual Army force structure with time phased unit requirements for a given scenario, identify overages and shortfalls on the force, create notional units for shortfalls, and produce movement requirements (e.g., origin destination, required delivery date, and deployment weight) as determined from the time phased unit requirements for input to various strategic mobility models. The CAMP provides the interface between the Force Accounting System (FAS) used in force structuring and analysis by the DA staff, the Force Analysis Simulation of Theater Administration and Logistics Support (FASTALS) Model used at CAA, and the SMOBSMOD and MORSA mobility models. Significant features of CAMP are its abilities to generate notional FAS records for required units not on the FAS file; to develop unit movement requirements for the notional units and for actual units which are required by the scenario; and to calculate nonunit movement requirements (resupply, replacements, and fillers) to support the force. The user may specify up to seven theaters, each with different force lists, and nonunit factors. In general, CAMP programs function sequentially. Should an error in input

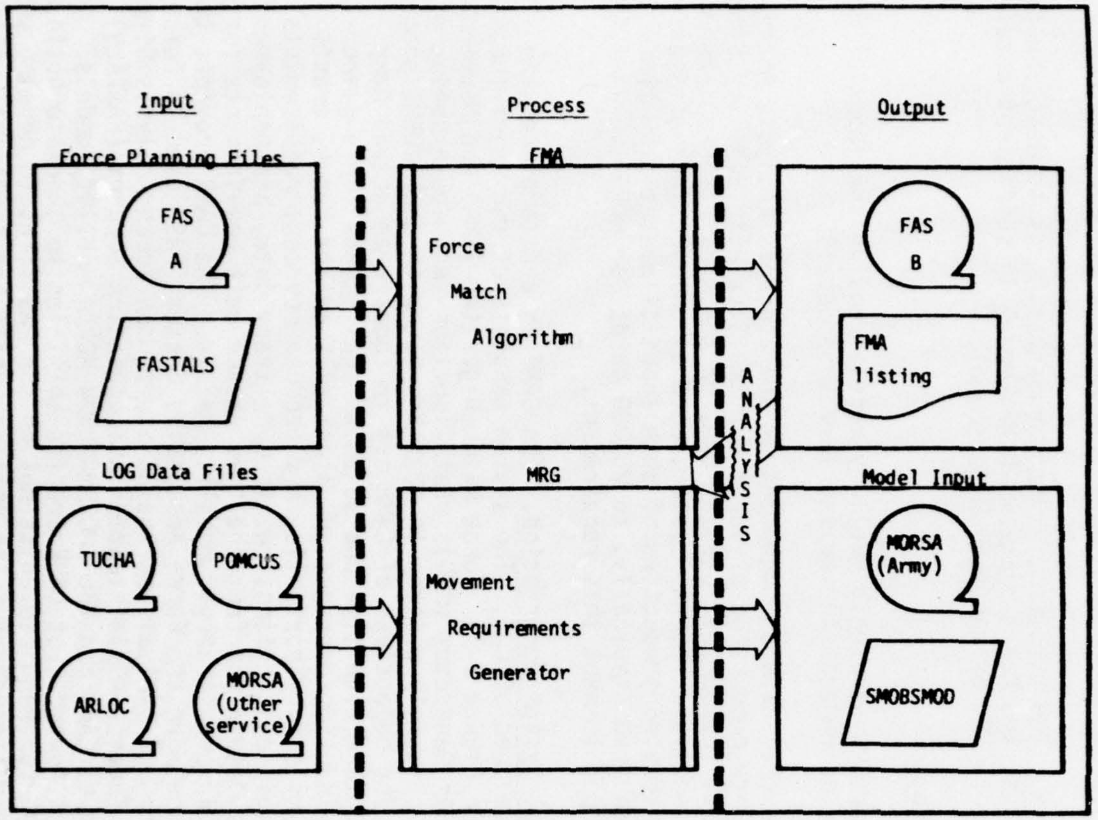


FIGURE I-1, CAMP Input-Processing-Output Chart

to one program lead to a processing error, execution is terminated and a diagnostic error message is printed to assist the user in correcting the error. The CAMP has been interfaced with the Unit Data System (UDS) to provide a generalized report generator capability.

4. Operation and Maintenance. The CAMP operates on the UNIVAC 1108 computer. The maximum program size is 58K words but most programs are between 20K and 40K words. Processing time is directly dependent on the size of the force being processed. Typically, a force may include on the order of 10,000 units, in which case exercise of the FMA will require approximately 1 hour of wall-clock time, and MRG will require approximately 2 hours. The CAMP is maintained by the CAA Methodology and Resources Directorate (MRD) that provides program-related user assistance. The proponent of CAMP is the Joint and Strategic Forces Directorate (JF), that provides functionally oriented user assistance.

5. Security. The CAMP programs are unclassified, but files and data are generally classified SECRET.

6. Limitations and Constraints

a. Data Base. Many of the files accessed by CAMP are not produced at CAA. The user often has little control over the quality of the data in these files. Close coordination with the data source is required to obtain valid data.

b. Formats. The input force must be in FAS file format. The output is produced only in SMOBSMOD and MORSA formats. Requirements for other formats should be coordinated with MRD.

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

CHAPTER II
FUNCTIONAL DESCRIPTION - FORCE MATCH ALGORITHM (FMA)

1. Introduction

a. The Force Match Algorithm (FMA) is a force structuring and analysis tool. The FMA compares an actual or planning force in FAS file format with a time phased requirements list produced by FASTALS. Selected units on the force may be locked out (excluded from consideration in the match) based on user input. For example, units located in a specific theater might be locked out in a match of requirements for another theater. The remaining units are match candidates and are matched against type unit requirements identified by FASTALS. Each match candidate is either matched or designated as excess to the stated requirements. Each requirement that cannot be matched with a unit on the force results in generation of a notional unit. Each matched unit and each notional unit are assigned a required delivery date (RDD) based on the FASTALS time phasing. Assignment of RDD within a FASTALS time period is user adjustable.

b. Output of the FMA is used by the force planner to allocate resources between active and reserve components. The rationale for allocation is that units required early should be active, while those required later may be reserve. Close scrutiny of overages and shortfalls is required to determine if substitutions can be made. It is from this manual analysis that CAMP draws the computer assisted portion of its name. The match cannot be fully automated because the force planner should have the final decision in structuring a force. However, CAMP output assists the planner in force analysis. In general, analysis by the planner will result in retention of many units designated as excess. Further, budgetary or other constraints may well result in the inability of the planner to activate the notional unit shortfalls. Only the application of sound judgment by the planner will result in designation of the appropriate force.

c. The FMA input consists of two basic files: the Force file and the Requirements file. The Force file is a planning force and may be considered as the starting point in the force structuring process. Typically, the Force file is provided in FAS format by the Office, Deputy Chief of Staff for Operations and Plans (ODCSOPS). The second input file is the Requirements file. This file is generated by FASTALS, the theater roundout model, in response to a specific scenario or situation.

2. System Processing Flow. The FMA consists of a series of nine programs that are identified in Table II-1.

a. Preprocessors

(1) Program BUILD accepts a FAS tape and creates a keyed access UDS file. The UDS allows detailed analysis of the force prior to match. In particular, the user can verify the number and location of various type units, examine the POMCUS packages, and update the file to correct errors or add new data. Capabilities of UDS are described in Reference 1.

(2) Program SORTUDS reads the UDS file. Records are sorted and written to a Force file (FF) on mass storage.

b. Main Programs. Input-processing-output charts for the main programs are shown in Figure II-1.

(1) Program FSORT reads the Force file to select Match Candidates. The user has considerable influence over the selection process. Certain fields of the FAS records can be updated based on user input. Subsequent to update, FSORT divides the force into five groups.

(a) Locked Out. These units are designated by the user based on the value of the ADC01 field. Locked Out units are excluded from consideration in the match process and are written to the Lock Out file (LOU). Generally, these will be units with specific missions in theaters not of interest in the current run. They are considered nondeployable.

(b) Alternate Theater. These units are also designated by the user based on the ADC01 field. Alternate Theater units are required to be deployed to theaters other than the primary theater. Requirements for these units are not generated by FASTALS. The units are excluded from match consideration but will ultimately be assigned RDD and destination in the specified theater; these units are written to the Alternate Theater file (ALT).

(c) Above-the Line. These are major combat forces (divisions and separate brigade sized units). Both the RDD and theater of these units are determined by the user. The Above-the-Line units are matched manually and are excluded from automated matching. These units are written to the Above-the-Line file (ATL).

(d) Hand Played Units. The user may specify a number of units (by UIC, ROBCO, or ADC01) to receive specific RDD. Hand Played Units will be excluded from match processing but will be

TABLE II-1, Force Match Algorithm Programs

Program	Language	Size (words)	Time (minutes)	Input files	Output files	Function
BUILD	COBOL	16 K	2	FAS tape	UDS file	Create keyed access UDS file
SORTUDS	COBOL	28 K	15	UDS file	FORCE	Load and sort FAS file
FSORT	FORTRAN	36 K	11	FORCE	LOU, HPU, ALT, CAN, ATL	Select and sort candidates
RSORT	FORTRAN	33 K	10	CAMPSUM	REQ	Load and sort requirements
MATCH	FORTRAN	35 K	30	REQ, CAN	MREQ	Match
ATL	FORTRAN	8 K	1	ATL	MATL	Set RDD on ATL units
ALT	FORTRAN	8 K	1	ALT	MALT	Set RDD on ALT units
L AYIN	FORTRAN	36 K	30	MREQ,HPU, LO, MALT, MATL	FOUT	Write match data on FAS
BINCOPY	COBOL	35 K	15	FOUT	FAS tape	Sort file and write FAS tape

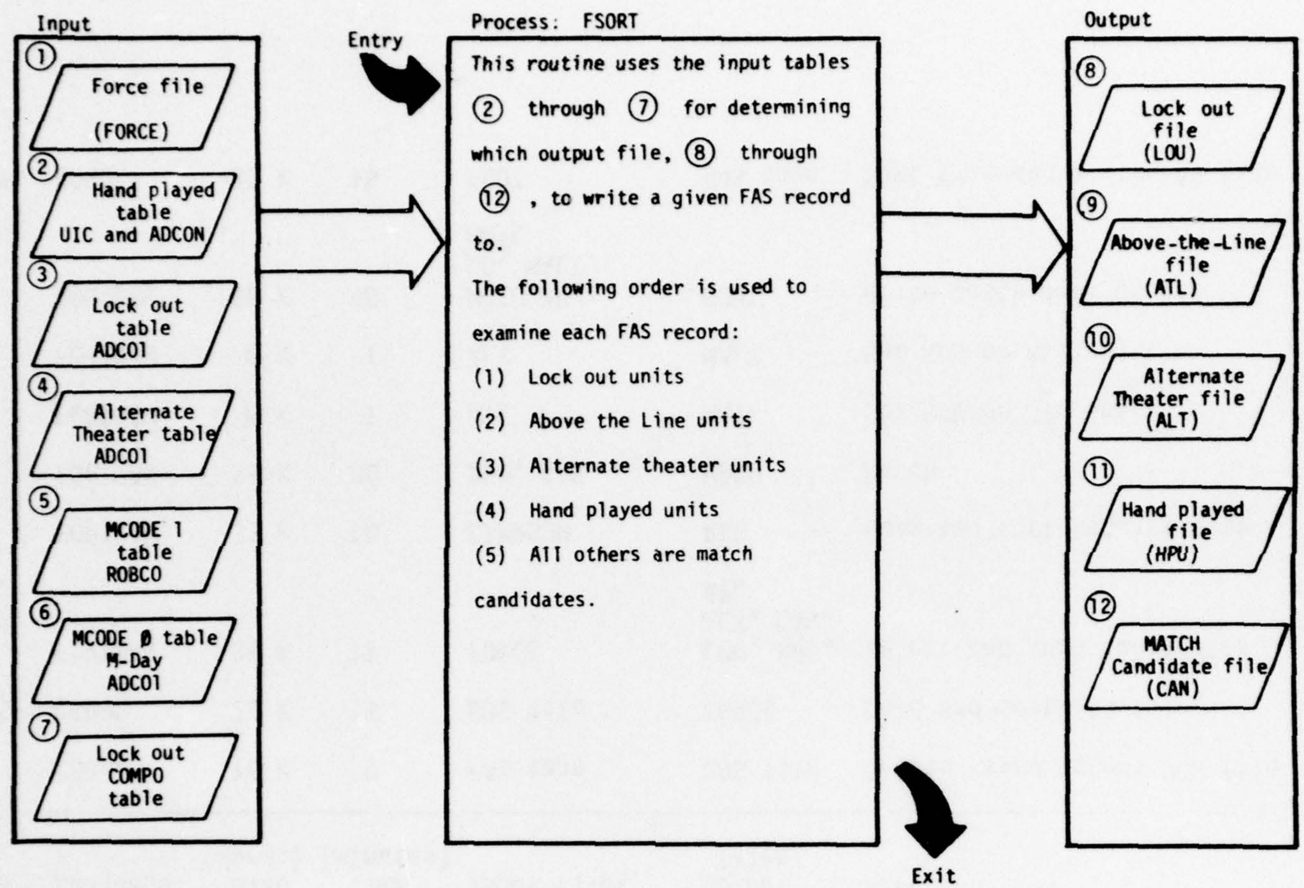


FIGURE II-1, FMA input-Processing-Output Chart (continued on next page) Page 1 of 6

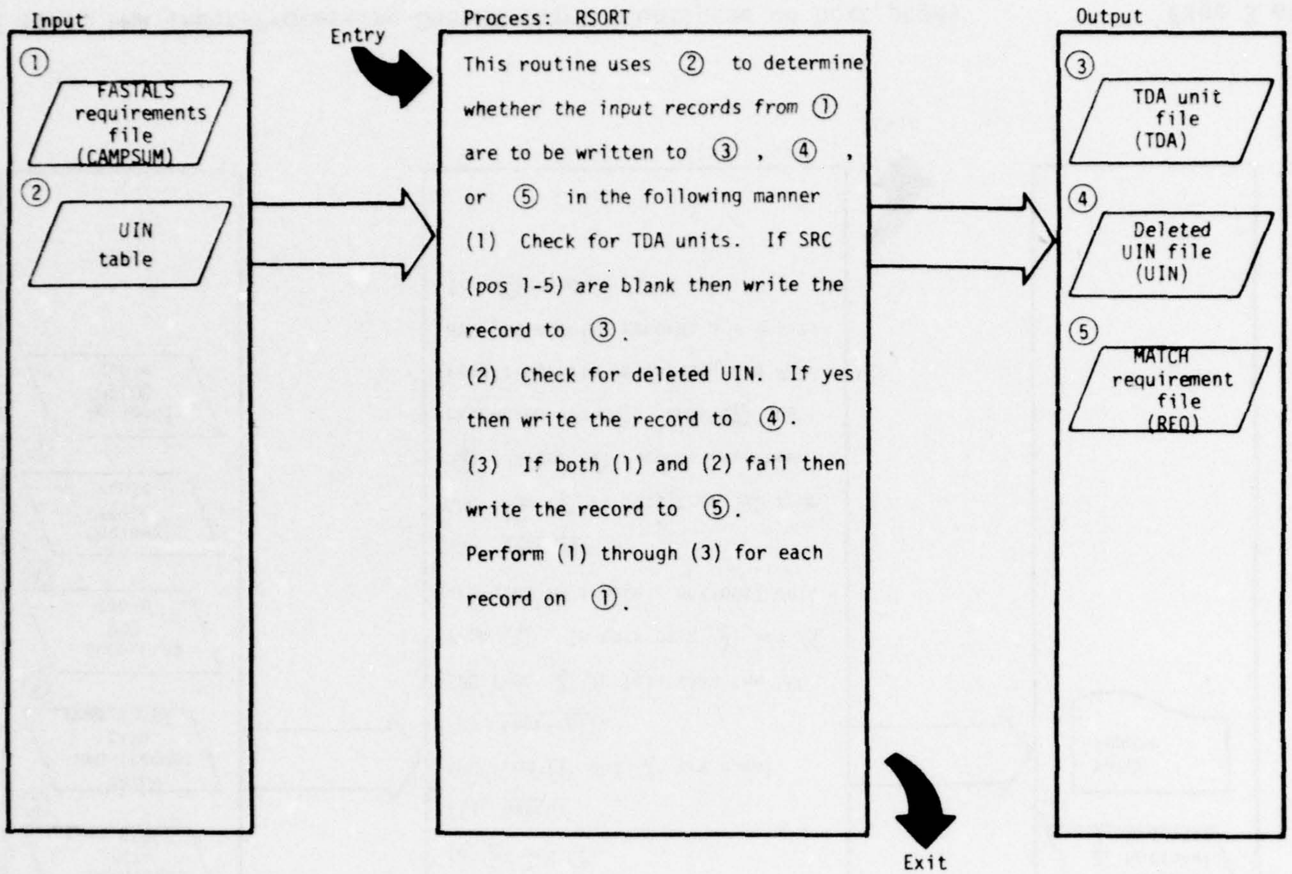


FIGURE II-1, FMA Input-Processing-Output Chart (continued on next page)

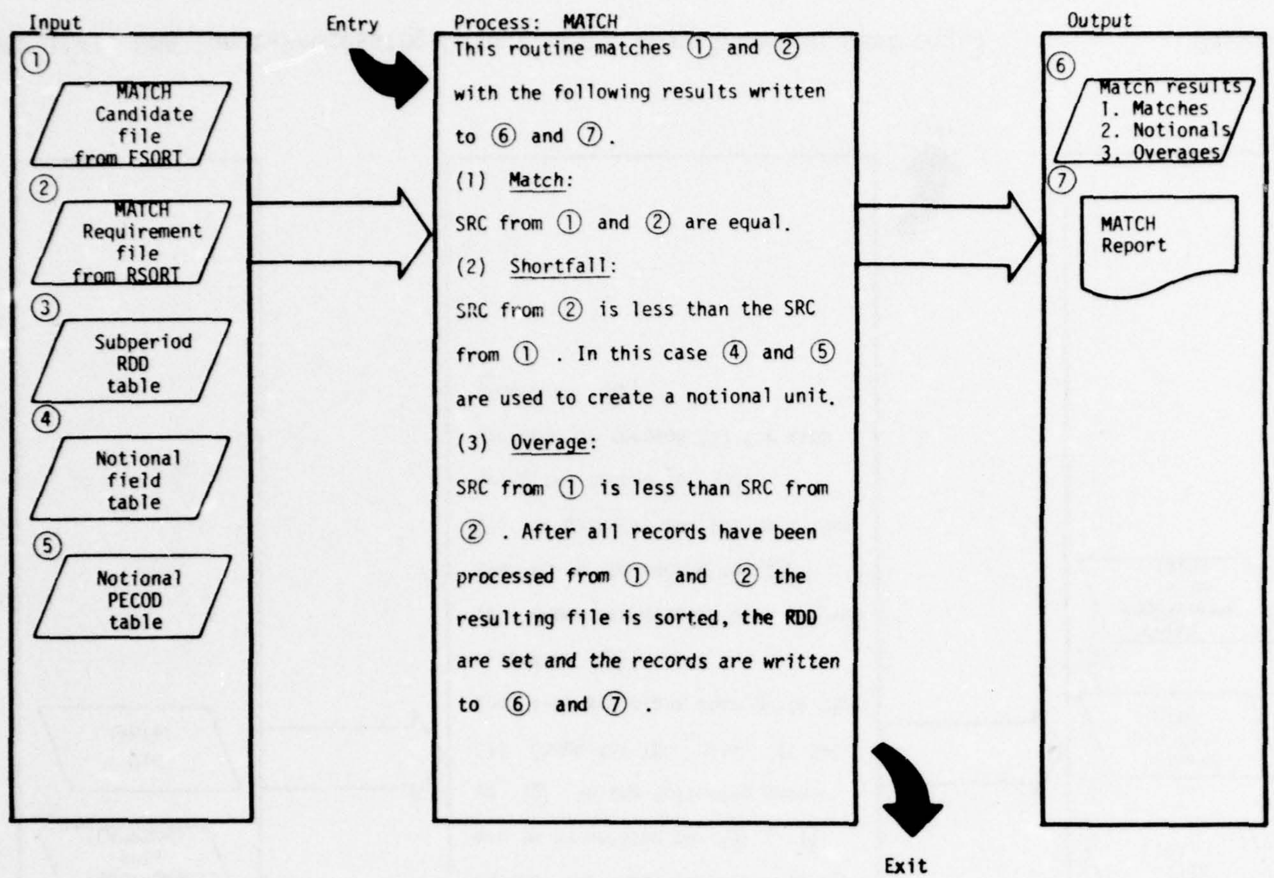


FIGURE II-1, FMA Input-Processing-Output Chart (continued on next page)

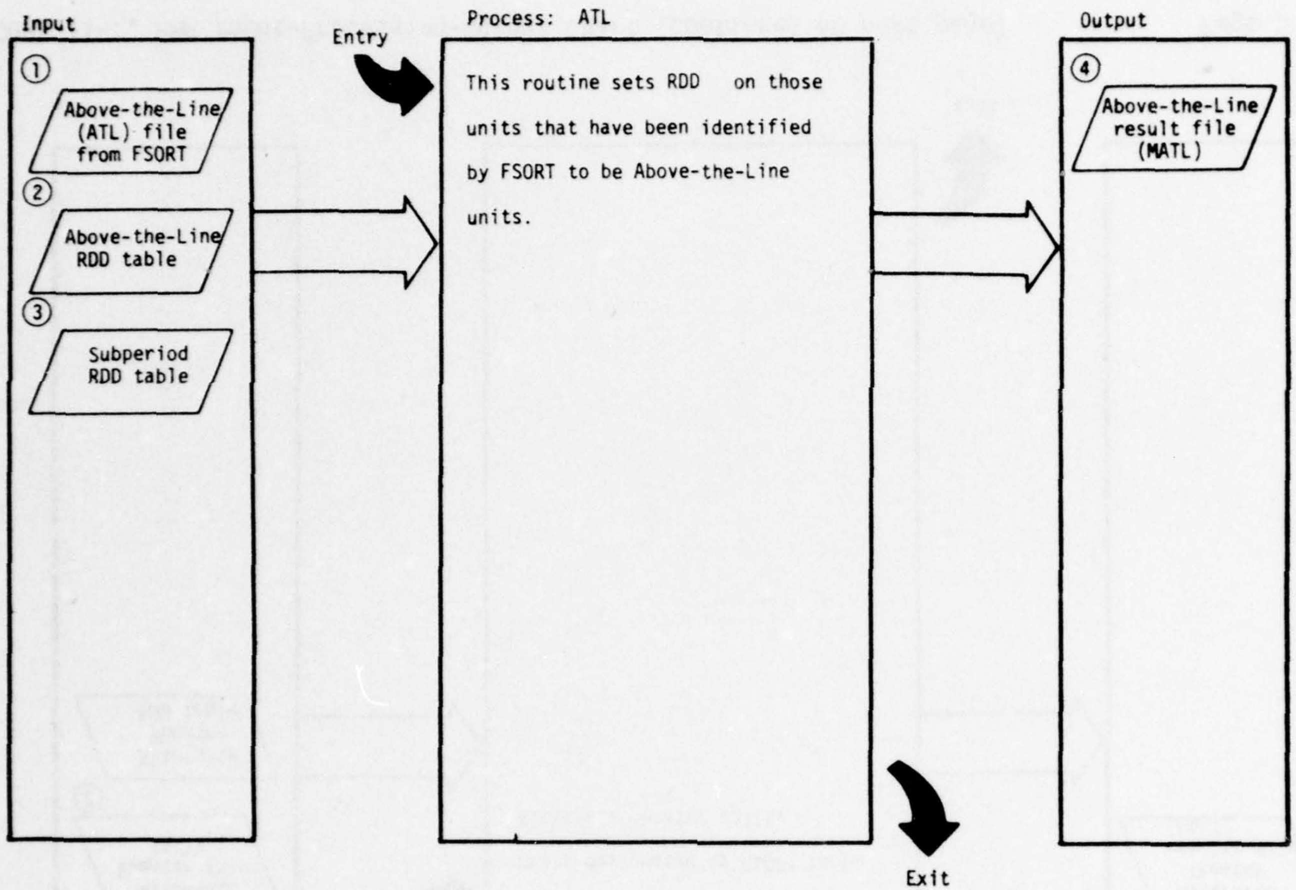


FIGURE II-1, FMA Input-Processing-Output Chart (continued on next page) Page 4 of 6

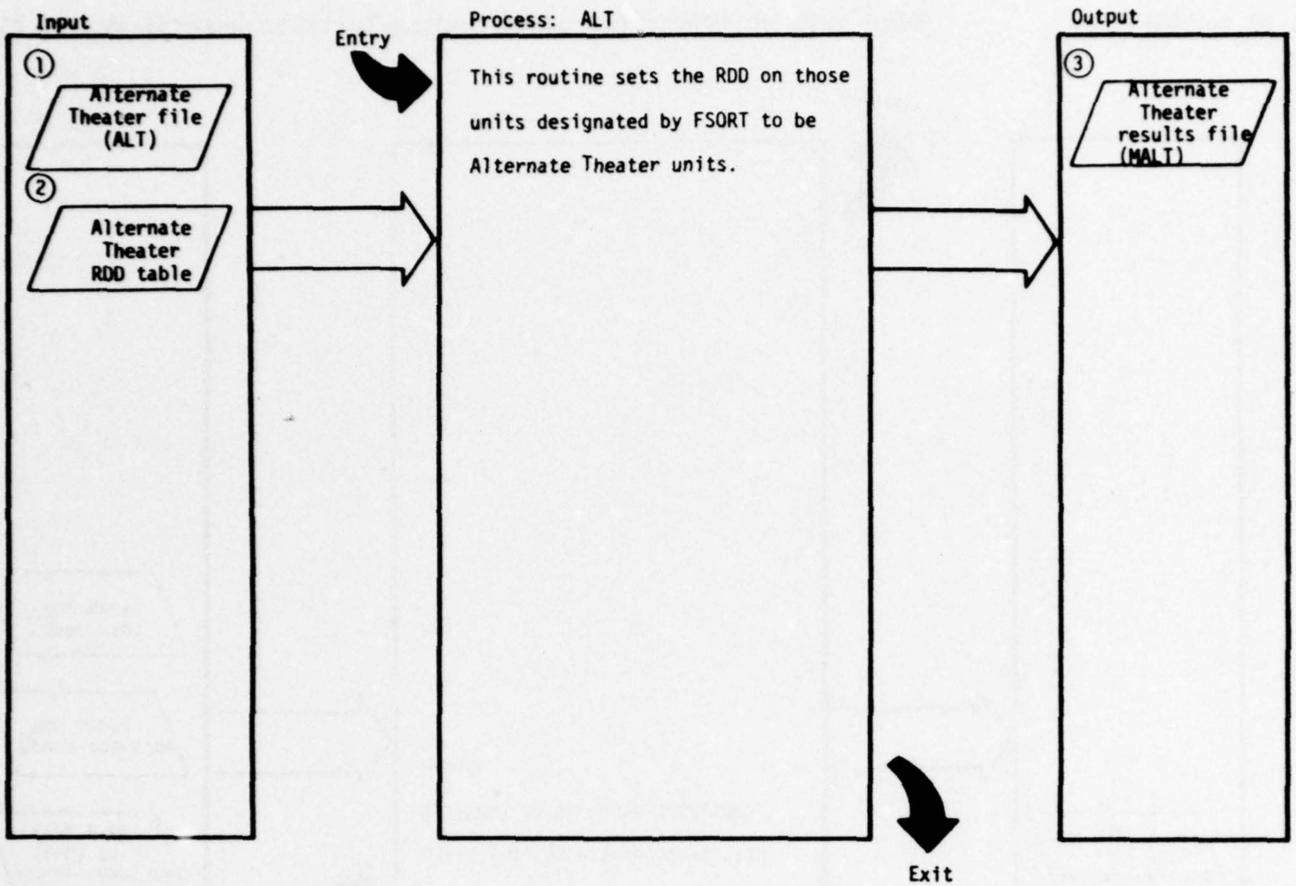


FIGURE II-1, FMA Input-Processing-Output Chart (continued on next page)

Page 5 of 6

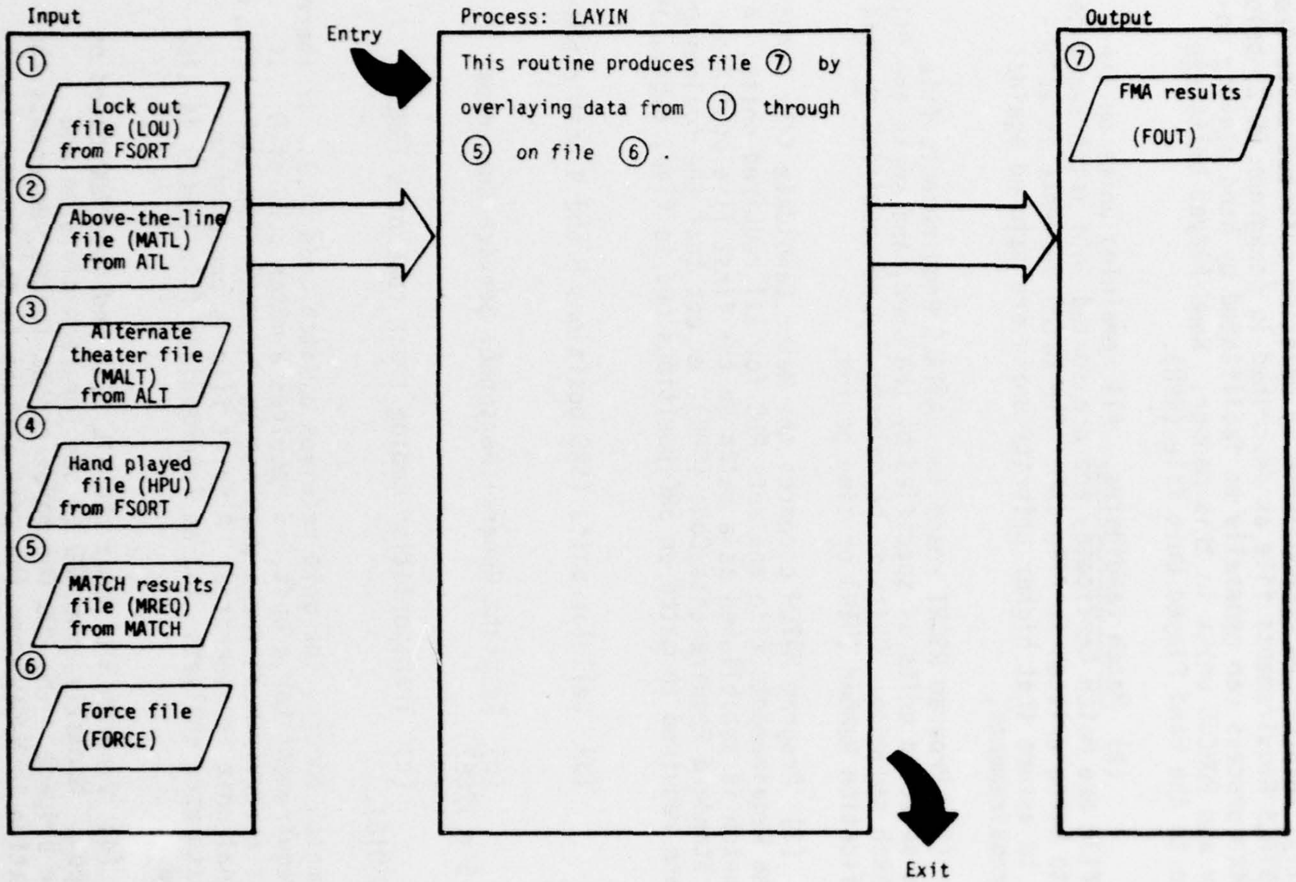


FIGURE II-1, FMA Input-Processing-Output Chart (concluded)

Page 6 of 6

designated as required in the primary theater, so the user must assure that duplicate requirements are not generated by FASTALS. Generally, this can be accomplished by deleting selected units from the FASTALS Requirements file as described in paragraph 2b(2) below. The match process can generally be facilitated by hand playing in-country and POMCUS units in this manner. Hand Played Units are written to the Hand Played Unit file (HPU).

(e) Match Candidates. All remaining units on the Force file are Match Candidates and are sorted into match sequence prior to being written to file CAN. The match sequence is established to assure that higher priority units are matched against early requirements.

(2) Program RSORT reads the FASTALS Requirements file, deletes selected units as specified by the user, and sorts the file into match sequence. Units to be deleted may be specified by Unit Identification Number (UIN) or time period.

(3) Program MATCH compares the Match Candidate file (CAN) with the Requirements file and sets RDD for all required units. A valid match is established as a match on the first five positions of the Standard Requirements Code (SRC), except that the following units are required to match on SRC positions one to five, eight, and nine.

(a) Cellular units (SRC positions 8 and 9 both alphabetic).

(b) Adjutant General Personnel Service Companies (SRC 1-5 = 12067).

(c) Transportation medium truck companies (SRC 1-5 = 55018).

When a match occurs, the unit receives a match code of 1. If there is no requirement for a unit, it receives a match code of 0. If there is a requirement for a unit not on the Match Candidate file, a notional unit is generated. A print file is produced to facilitate user analysis, and an intermediate file (MREQ) is also produced.

(4) Program ATL reads the ATL file and sets RDD based on user input. Normally, one RDD is set for each division and separate brigade. The RDD for major combat forces represents the day relative to M-day when the unit arrives in the theater (port discharge date). This date is usually established based on input to a warfighting model. The time period is also set. Output file is file MATL.

(5) Program ALT sets RDD for units on file ALT based on user input. Output is file MALT.

(6) Program LAYIN merges the files LOU, MALT, MATL, HPU, and MREQ with the Force file to record the match results. A revised Force file is developed with match results overlaid on selected fields:

(a) AUTH 1-3 = FASTALS UIN for matched or notional unit
= 000, overage
= blank, otherwise

(b) AUTH 4-7 = FASTALS rules for matched or notional unit
= zero, otherwise

(c) AUTH 8 = Match code, assigned as follows:

Required unit: 1
Not required: 0
Other, as specified for specific run (FAS ADC01)

(d) AUTH 9-10 = FASTALS time period for Required unit
= 99, overage
= blank, Lock Out or Alternate Theater

(e) AUTH 11-13 = RDD for Required unit or Alternate Theater
= 999, overage
= blank, Lock Out

(f) AUTH 14 = FASTALS Logical Region of Employment (LRE) for Required units
= zero for overage
= blank, otherwise

(g) AUTH 15 = Theater code as assigned by Match.

c. Postprocessors

(1) Program BINCOPY reads and sorts the revised Force file (FOUT) to produce the final output tape in FAS format. If the primary tape is to be forwarded to the DA staff for analysis, the user should make a copy for retention at CAA.

(2) Program BUILD may again be used to interface CAMP with the Unit Data System (UDS) by creating a keyed access UDS file from the FAS tape. The UDS capabilities are described in Reference 1.

3. Interfaces with other Models. Much of CAMP utility results from its capability to interface with other models and systems. The FMA interfaces with three major models to provide match results. Requirements data are extracted from FASTALS; forces are extracted from FAS; output is fed back to FAS and into UDS for analysis.

a. FASTALS Interface. The FASTALS generates time phased support requirements for a given tactical scenario. A special program called CAMPSUM, written and maintained by the CAA Methodology and Resources Directorate, produces CAMP input from FASTALS output. Actually, there are three versions of CAMPSUM, each of which handles composite units (called '66' units because the SRC ends in '66') differently. Composite units are composed of a mix of teams and detachments and are tailored to meet specific requirements. There is no predetermined number or type of teams that make up a specific composite unit. However, FASTALS plays "type" composite units. The principal version of CAMPSUM is called 66 ROLLUP because the teams and detachments are rolled up to the level of the '66' SRC. In general, composite units on the FAS file do not have a '66' SRC but instead display the SRC of the header team. To facilitate matching, a new version of CAMPSUM (66 ROLLUP--MODIFIED) was developed to roll up composite units using total strengths but using the SRC of the header team. A third version (66 ROLLOUT) displays requirements for each team and detachment without using the composite unit designation. This version must be used if movement requirements are to be generated. CAMPSUM also accesses the table(s) of organization and equipment/troop program sequence number (TOE/PSN) file to append detail strength data and the TPSN to the FASTALS output.

b. FAS Interface. The FAS interface is the weakest link of CAMP. Concepts Analysis Agency does not control the FAS data base. The interface is in the form of manual transfer of tapes between ODCSOPS and CAA. Errors in FAS data create significant problems for CAMP. Three alternatives are open to the user to correct FAS data errors. A new tape can be requested from ODCSOPS, the existing tape can be loaded into UDS and updated by the user, or program SORTUDS can be revised to make the required data changes. The best alternative to choose depends on the time available and the nature of the changes.

c. UDS Interface. The UDS provides two major capabilities: the data base update capability discussed above and a generalized

report generator capability. The interface is provided by program BUILD. The UDS reports are of great value to the CAMP user in analyzing match results.

4. User Constraints

a. Data Base. A significant detractor of FMA utility is the lack of control over the FAS data base. Close coordination with ODCSOPS can eliminate many potential errors and correct errors that have already occurred.

b. SRC Substitutions. The SRC used in FASTALS do not correspond exactly with those used by FAS. Apparent mismatches result in generation of unnecessary notional units when in fact a similar unit is on the force but has a different SRC.

c. Composite Units. Actual composite units on the FAS file are tailored to meet specific requirements. The composition of such units may differ greatly from the type composition used by FASTALS. Thus, a given SRC may represent one type unit in FAS and another type unit in FASTALS. The best way to resolve this is by rolling out composite units to the team level. Efforts are underway to achieve this capability on the FAS file.

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

CHAPTER III
FUNCTIONAL DESCRIPTION - MOVEMENT REQUIREMENTS GENERATOR (MRG)

1. Introduction. The Movement Requirements Generator (MRG) interfaces FMA output with selected strategic mobility models. The MRG automates the production of input data for Strategic Mobility Simulation Model (SMOBSMOD) and the Mobility Requirements for Staff Analysis (MORSA) system. The MRG determines unit movement requirements for a force to be deployed. The force is entered in FAS format. The force may be entered directly from FMA; however, the more general occurrence is that FMA output is analyzed and reviewed by the DA staff prior to being entered in MRG. Actual unit locations are used as origins, and location codes required by the models are obtained automatically through an MRG interface with the Army Location file. Deployment weights for various categories of unit equipment are obtained automatically from the TUCHA file. A special file is used for those units that have equipment prepositioned in the theater. Accompanying supply weights are entered by the user. Nonunit movement requirements (fillers, replacement, and resupply) are calculated by MRG based on factors entered by the user. These factors include consumption rates, prepositioned quantities, and theater stockage objectives. After all movement requirements have been determined, the data are reformatted to meet input specifications for SMOBSMOD and MORSA. An automated interface allows the data (both unit and nonunit) to be entered into UDS, thus facilitating review and analysis.

2. System Processing Flow. The MRG consists of 13 programs as listed in Table III-1. Processing logic is shown in Figure III-1 and is discussed below.

a. Programs SORTARLOC, MORSAROLL, TUCHAROLL, and POMCUSROLL are preliminary processes that build the data base actually used in subsequent MRG processes. Each of these programs reads a tape produced outside of CAA and creates a mass storage file with data required for subsequent processing. The formats for the tapes and output files are described in subsequent chapters. Use of these preliminary processes allows the user to analyze the input data thoroughly prior to making production runs. The computer resources required for production runs are thus considerably reduced because the output files are generally smaller than the input files and operate on time-saving mass storage files in lieu of tape files. In MORSAROLL, the user can specify how other service movement requirements are to be rolled up for SMOBSMOD. Origins and destinations may be rolled to specific SMOBSMOD nodes, and RDD and availability dates may be rolled into groups.

TABLE III-1, Movements Requirements Generator Programs

Program	Language	Size (words)	Time (minutes)	Function
SORTARLOC	COBOL	18 K	5	Produces Location file
MORSAROLL	COBOL	36 K	6	Rolls up other service movement data for SMOBSMOD
TUCHAROLL	COBOL	42 K	6	Produces TUCHA summary file
POMCUSROLL	COBOL	42 K	2	Produces POMCUS summary file
GEO	FORTRAN	58 K	16	Sets origin and destination codes
LOG	FORTRAN	34 K	28	Sets unit deployment weights. Produces MORSA cards
UPMDF1	FORTRAN	20 K	20	Allows SRC substitution
NUR	FORTRAN	38 K	11	Calculates nonunit movement requirements. Produces MORSA cards
PCKAGE	FORTRAN	36 K	14	Rolls movement requirements into SMOBSMOD packages
PRTPCK	FORTRAN	6 K	2	Prints report
MORSAMERGE	FORTRAN	6 K	12	Merges unit and nonunit MORSA cards
INLAY	FORTRAN	37 K	25	Lays movement data on FAS records
BUILDMRG	COBOL	16 K	2	Produces keyed access UDS file

Input files	Processors	Output files
MORSA tape (other service) MORSAROLL data	MORSAROLL	SMOBSMOD input tape OSV (other service data)
TUCHA tape	TUCHA ROLL	28-TUSUM file
POMCUS tape	POMCUS ROLL	27-POMSUM file
Army location file tape	SORT ARLOC	15-GEO loc file
15-GEO loc file from SORTARLOC 17-FAS file a/ GEO data Node data	GEO	29-MDF1 (movement data file) 7-Print of MDF1

a/Normally produced by FMA program LAYIN. May be produced by SORTUDS if no match processing is required.

FIGURE III-1, MRG Input-Processing-Output Chart (continued on next page)

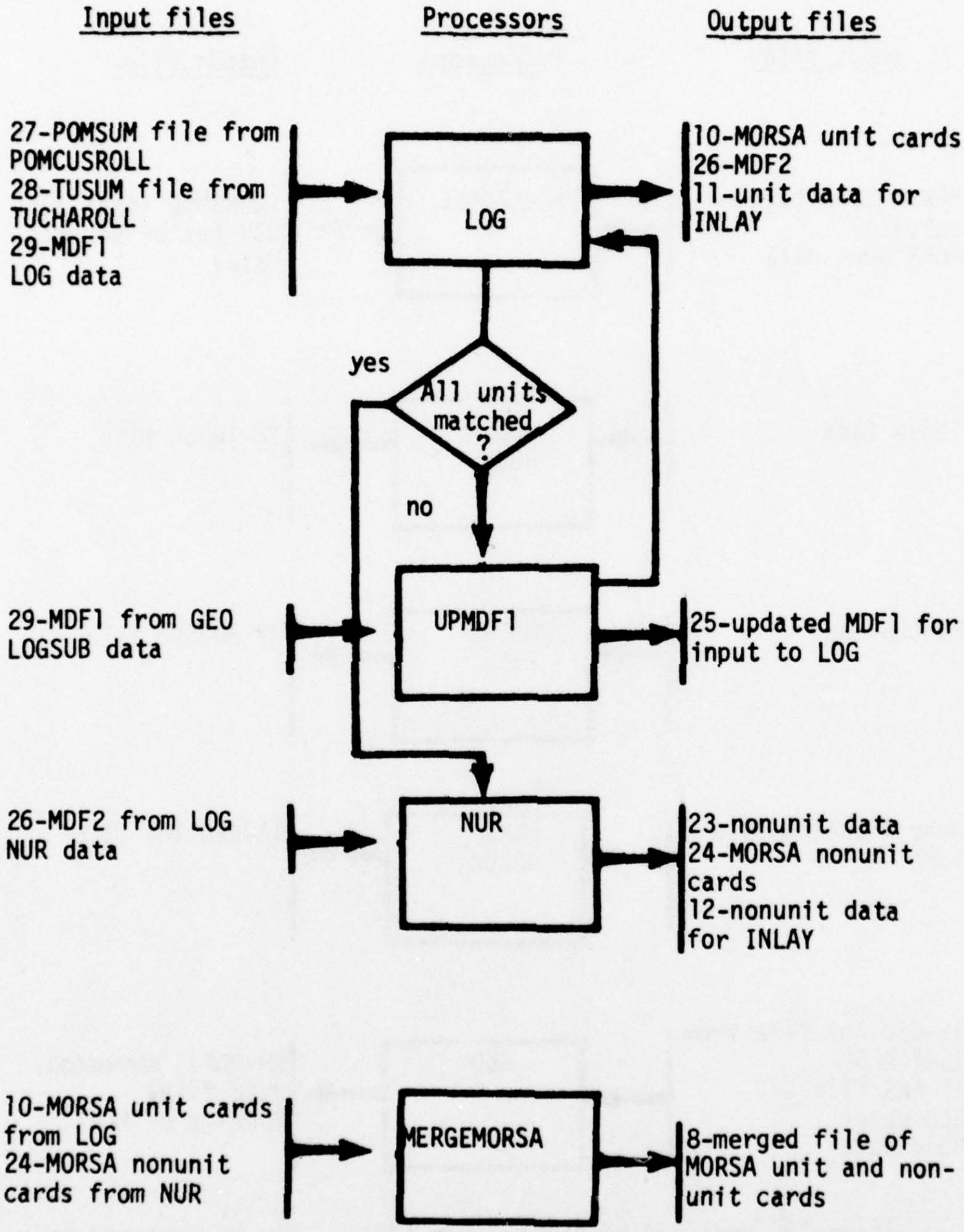


FIGURE III-1, MRG Input-Processing-Output Chart (continued on next page)

Input files	Processors	Output files
24 OSV (other service data) from MORSAROLL 23-nonunit data from NUR 26-MDF2 from LOG PCKAGE data	PCKAGE	19-nonunit package data 20-print 1 line per package 21-unit package data 22-SMOBSMOD cards
19-nonunit package data from PCKAGE 21-unit data from PCKAGE	PRTPCK	print of packages
17-FAS file a/ 11-unit data from LOG 12-nonunit data from NUR INLAY data	INLAY	FAS tape
FAS tape from INLAY	BUILDMRG	UDS file

---

a/Normally produced by FMA program LAYIN. May be produced by SORTUDS if no match processing is required.

FIGURE III-1, MRG Input-Processing-Output Chart (concluded)

page 3 of 3

b. Program GEO accepts as input the FAS file (normally produced by FMA program LAYIN), the GEO Location file (produced by SORTARLOC), a Node Table entered by the user, and various data cards entered by the user. Program GEO processes each unit on the FAS file. If the unit is not to be deployed to a designated theater (entered by user), it is ignored. If the unit is to be deployed to a designated theater, GEO searches the Location file to determine the origin geographic location code (GELOC) and the Node Table to determine the SMOBSMOD node number. The destination GELOC and node and the availability date are set based on user input for each theater. Travel mode is determined. The user may enter a list of exceptions, based on Troop Program Sequence Number (TPSN), that override normal program logic to set selected data items. Program GEO produces the Movement Data File 1 (MDF1) that identifies each unit movement requirement.

c. Program LOG accesses the TUCHA summary file (TUSUM) produced by TUCHAROLL, the POMCUS summary file (POMSUM) produced by POMCUS-ROLL, and accompanying supply data entered by the user to obtain deployment weights for units on file MDF1. If a unit designated as POMCUS on MDF1 is not found on the POMCUS summary file, it is assigned zero weight. If any SRC on MDF1 are not found on the TUCHA summary file, error messages are printed out to the user. The user then runs program UPMDF1 to substitute valid SRC for those not found, and subsequently reruns program LOG. If all SRC are found, program LOG produces three files. Movement Data File 2 (MDF2) includes all data in MDF1 plus the deployment weights in each of 17 cargo categories; this is used in PCKAGE to generate SMOBSMOD input. The MORSA unit card images are written out to a separate file and a third file contains unit data for program INLAY.

d. Program NUR calculates nonunit movement requirements.

(1) Fillers represent personnel required to bring in-country units to full strength. Fillers are calculated based on user input.

(2) Replacements are calculated at 5-day increments based on casualty data entered by the user and the theater strength, which is determined dynamically based on unit strength and RDD information in MDF2 file.

(3) Resupply requirements are calculated for POL, ammunition, and dry bulk. For each theater, the user enters prepositioned quantities, consumption data, and theater stockage policies (day resupply starts, RDAY; stockage objective; and day objective is to be achieved, BLDUP). Program NUR simulates arrival of units and the arrival and consumption of supplies to determine resupply requirements. For all days prior to RDAY, the model generates

resupply requirements to maintain a safety level (entered by the user). For each 5-day interval between RDAY and BLDUP, the model interpolates between the on-hand level at RDAY and the stockage objective at BLDUP to determine requirements. However, if RDAY level is greater (in terms of days of supply) than the stockage objective, resupply requirements are not generated until the level falls below the objective. For each 5-day increment after BLDUP, the model generates resupply requirements to maintain the stockage objective. Nonunit requirements are written to a MORSA card file and a file containing data for program INLAY.

e. Program MERGEMORSA produces a single set of MORSA card images on tape by merging the MORSA files produced by LOG and NUR.

f. Programs PCKAGE and PRTPCK produce SMOBSMOD input cards and print out the results for user review. Both a detailed and a summary report are produced. Movement requirements are rolled into packages. All requirements with the same RDD, availability date, origin node, destination node, and travel mode are rolled into one package. The user may obtain a higher degree of rollup by specifying input parameters that cause several RDD and several availability days to be packaged together such that the package will have the earliest RDD and latest availability day of the group.

g. Program INLAY creates a FAS file tape overlaid with movement data. New FAS records are created for nonunit requirements; the user enters the FICOD, COMPO and EDATE, and the UIC (first position N) is set sequentially by INLAY. Units on the FAS file that are not deployed are not overlaid with movement data.

h. Program BUILDMRG reads the FAS tape created by INLAY and produces a keyed access UDS file.

3. Interfaces with Other Models. The MRG is designed specifically to interface the FAS with SMOBSMOD and MORSA. Interfaces with additional mobility models could be programed if required by the user and if all required data items are available in CAMP files. Requirements for additional interfaces should be coordinated with the CAA Methodology and Resources Directorate. The MRG also interfaces with FAS and UDS.

a. The MRG/SMOBSMOD interface provided by program PCKAGE results in production of SMOBSMOD input card images in a mass storage file. The SMOBSMOD input requirements are described in Chapter VI and overall SMOBSMOD capabilities are described in Reference 2.

b. The MRG/MORSA interface results in production of MORSA

input card images on tape. Concepts Analysis Agency does not currently have the capability to process MORSA data, but is required by various study directives to produce the data for processing at the Joint Chiefs of Staff (JCS) level. Input to MORSA is described in Chapter VI and is produced by programs LOG and NUR and merged by program MERGEMORSA. The MORSA system is operated and maintained at the Command and Control Technical Center (CCTC), Logistics Data System Division, and is somewhat volatile. The user is cautioned to coordinate with CCTC prior to running MRG to ascertain if any MORSA input requirements have changed. The MORSA system is described in Reference 3.

c. The MRG/UDS interface provided by program INLAY results in a significant report generation capability. The UDS file as created by MRG is described in Chapter VI. Overall UDS capabilities are described in Reference 1.

d. An MRG/FAS interface can be provided by program SORTUDS to allow use of an actual force (with or without notional units) as a basis for determination of movement requirements without prior FMA processing. In this case, the force must contain unit RDD in the ADC03 field.

4. User Constraints

a. Data Base. Inconsistencies occur relatively frequently in the various files accessed by MRG. The user should review the data contained in all computer tapes obtained outside of CAA. Errors and inconsistencies should be coordinated with the source of the data.

b. MORSA Output. Concepts Analysis Agency does not have the capability to process MORSA data. The MORSA system is operated by CCTC on the JCS Worldwide Military Command and Control System (WWMCCS) computer, and CAA is required to provide movement data in MORSA format. However, correction of errors is difficult and often requires that CAA rerun the job.

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

CHAPTER IV

DATA DICTIONARY

Data element	Description	Set	Used	Remarks
AØ	Weight of non-air transportable (NAT) vehicles.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	TUCHAROLL POMCUSROLL LOG	Rolled up for output to MORSA. Output to SMOBSMOD cargo category 2.
A1	Weight of outsize vehicles.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	TUCHAROLL POMCUSROLL LOG	Rolled up for output to MORSA. Output to SMOBSMOD cargo category 5.
A2C	Weight of oversize containerizable vehicles.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	TUCHAROLL POMCUSROLL LOG	Rolled up for output to MORSA. Output to SMOBSMOD cargo category 10.
A2D	Weight of oversize noncontainerizable vehicles.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	TUCHAROLL POMCUSROLL LOG	Rolled up for output to MORSA. Output to SMOBSMOD cargo category 11.
A3C	Weight of containerizable bulk vehicles.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	TUCHAROLL POMCUSROLL LOG	Rolled up for output to MORSA. Output to SMOBSMOD cargo category 14.

Data element	Description	Set	Used	Remarks
ACAMO	Weight of accompanying ammunition.	LOG	NUR	Output to MORSA (short tons) and SMOBSMOD (category 17)(thousand short tons).
ACTCO	Action code. See data element EDATE.	Input from FAS file.		Identifies the type change being made on a given EDATE. E.g., ACTCO = J implies deactivation of a unit.
ADCON				
ADC01	See data element Theater Code.			
ADC02	See data element Time Period.			
ADC03	See data element RDD.			
AMMO	Ammunition weight.	NUR for resupply.		Output to MORSA and SMOBSMOD (category 17).
AREA	Area abbreviation. A three-position code which identifies a specific geographic area. See data elements LOCCO and MBLOC.	Input from Army location file.	GEO	DA Cir 525-10-4, -3. CONUS areas are identified by Army area and state. Example: 1VA implies First Army, Virginia. Overseas areas are generally identified by an abbreviation of the country name. Example: FRA for France.

Data element	Description	Set	Used	Remarks
ARLOC	Army location code. A unique five position code which identifies a geographic location at which Army activities may be situated or which may have potential military significance.	Input from Army location file.	SORTARLOC	
ASGMT	Assignment. Identifies major command to which a unit is assigned.	Input from FAS file.		Manually set for notional units.
ASPLY	Weight of accompanying supply (excluding ammunition).	NUR		Output to MORSA and SMOBSMOD.
AUTHR				
AUTHR 1-3	See data element UIN.			
AUTHR 4-7	See data element Rules.			
AUTHR 8	See data element Match Code.			
AUTHR 9-10	See data element Time Period.			
AUTHR 11-13	See data element RDD.			

Data element	Description	Set	Used	Remarks
AUTHR 14	See data element LRE.			
AUTHR 15	See data element Theater Code.			
AVAIL	Availability date. Day relative to the starting day of a scenario on which a given unit is available for deployment.	GEO		Output to MORSA and SMOBSMOD.
BØ	Square feet of NAT non-self deployable aircraft (NSDA).	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC.		Rolled up for output to MORSA. Output to SMOBSMOD category 3.
B1	Square feet of outsize NSDA.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC.		Rolled up for output to MORSA. Output to SMOBSMOD category 4.
B2C	Square feet of containerizable oversize NSDA.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC.		Rolled up for output to MORSA. Output to SMOBSMOD category 8.
Branch	Branch of service for TOE units.	Input from FAS file and FASTALS.	N/A	Example: EN for engineers.

Data element	Description	Set	Used	Remarks
BULK	Cargo having dimensions not larger than 104 x 84 x 96 inches.	LOG		Output to MORSA.
CARRS	Combat arms regimental system. Historical designation assigned to combat arms units.	Input from FAS file.	N/A	
COMPO	Component code. Identifies unit status.	Input from FAS file.	FSORT	1 = Active 2 = National Guard 3 = Reserve 4 = Notional.
DAMPL	DA master priority list. Priority grouping of units for allocation of personnel and/or equipment.	Input from FAS file.	FSORT	Match sequence is keyed on DAMPL.
DSCMP	Display/compute indicator.	Input from FAS file.	SORTUDS	DO implies display only (a titular record). DC implies display and compute (i.e. count this unit's strength).
Destination	The location to which a unit is being deployed.	GEO. Based on Theater Code.	NUR	Output to MORSA and SMOBSMOD.

Data element	Description	Set	Used	Remarks
Dry Re-supply	Quantity of resupply in all classes except III (POL) and V(ammo) (thousands of tons).	NUR		SMOBSMOD cargo category 16.
EDATE	Effective date of a transaction. See data element ACTCO.	Input from FAS file.		Manually set for notional units.
FCTCD	Foreign Country Code Description. The full name of foreign countries.	Input from Army location file.	N/A	DA Cir 525-10-4.
FICOD	Force identification code. Differentiates between various planning forces.	Input from FAS file.		Example: J = JSOP force T = Total Force.
FPLAN	Force Planning Code. Classification code for strategic and force planning.	Input from FAS file.	N/A	First position: A = Division forces B = Special mission C = General support.
GELOC	Specified geographic location code. A JCS established equivalent to the ARLOC covering locations for all military activities.	Input from Army location file.	GEO	Output to MORSA.

Data element	Description	Set	Used	Remarks
J1	Quantity of non-vehicular outsize cargo.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	LOG	Rolled up for output to MORSA (short tons). Output to SMOBSMOD category 7 (thousand short tons).
J2C	Quantity of non-vehicular oversize containerizable cargo.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	LOG	Rolled up for output to MORSA (short tons). Output to SMOBSMOD category 13 (thousand short tons).
J2D	Quantity of non-vehicular oversize noncontainerizable cargo.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	LOG	Rolled up for output to MORSA (short tons). Output to SMOBSMOD category 12 (thousand short tons).
J3C	Quantity of non-vehicular bulk containerizable cargo.	Input from TUCHA file for each SRC. Input from POMCUS file for each POMCUS UIC (tenths of short tons).	LOG	Rolled up for output to MORSA (short tons). Output to SMOBSMOD category 15 (thousand short tons).
J7C	Quantity of accompanying dry bulk supply.	Input from TUCHA file.	N/A	
JCSTY	JCS type unit code. JCS designated equivalent of SRC.	Input from FAS and TUCHA files.	N/A	Output to MORSA.

Data element	Description	Set	Used	Remarks
Level	First three characters of the FAS unit description. Denotes the organizational level of the unit.	Input from FAS and FASTALS files.	N/A	Example: TM = team BN = battalion Note: Not to be confused with level at position 10 of SRC.
LFLLC	Geographic coordinates. Latitude and longitude.	Input from Army location file.	N/A	
LOCCO	See data element AREA. The area code for the unit home station.	Input from FAS file.	GEO	
LOCNA	Location name abbreviated. Nine characters maximum.	Input from Army location file.	GEO	
LOCNM	Location name; 17 characters maximum.	Input from Army location file.	GEO	
LRE	Logical Region of Employment.	Input from FASTALS.		1 = combat zone 2 = corps forward 3 = corps rear 4 - commz forward 5 = commz rear 6 = offshore Output to FAS.
Match Code	Indicator for match results.	MATCH	LAYIN	1 = required 0 = excess X = locked out.

Data element	Description	Set	Used	Remarks
MBLOC	See data element AREA. Area code for unit mobilization station.	Input from FAS file.	GEO	
MBSTA	See data element LOCNA. Abbreviated location name of unit mobilization station.	Input from FAS file.	GEO	
NAT	Non-air transportable. Cargo having dimensions larger than 1453 x 144 x 156 inches or 1453 x 216 x 114 inches.	LOG		Output to MORSA.
NSDA	Non-self-deployable aircraft.	LOG		Output to MORSA.
Origin	Location at which a unit or package originates deployment.	GEO. Usually home station for active units, mobilization station for nonactive units.		Output to MORSA as the GELOC. Output to SMOBSMOD as a node number.
OUTSZ	Outsize. Cargo having dimensions larger than 810 x 117 x 105 inches but not excluding the limits of air transportability.	LOG		Requires C5A aircraft. Output to MORSA.

Data element	Description	Set	Used	Remarks
OVRSZ	Oversize. Cargo having dimensions larger than 104 x 84 x 96 inches but no larger than 810 x 117 x 105 inches.	LOG		Requires C-141 aircraft. Output to MORSA.
PAX	Passengers.	Input from TUCHA.	NUR LOG	Output to MORSA and SMOBSMOD.
PECOD	Program element code. Budget code which groups similar forces.	Input from FAS file.		Manually set for notional units.
PHASE	Authority for a FAS record entry.	Input from FAS file.		Manually set for notional units.
RD	Record description. Code to identify the type cargo. See data element RT.	NUR LOG		Output to MORSA. A = ammo F = fillers G = resupply dry or replacements N = NSDA 8 = POL.
RDD	Required delivery date. Day relative to initial day of a scenario when unit or package is required to arrive in the destination theater (port discharge date).	MATCH ATL ALT	NUR	Output to MORSA and SMOBSMOD.

Data element	Description	Set	Used	Remarks
ROBCO	Ostensibly, a code to identify reserve component units or part of a force grouping. Now used extensively to designate POMCUS packages.	Input from FAS file.	FSORT LOG	Output to MORSA.
RT	Record type code to identify the type cargo. See data element RD.	NUR LOG		Output to MORSA U = unit P = POL S = other.
Rules	Zero/one variables to indicate reason for a unit requirement. Manual play, theater structure, existence and/or workload.	Input from FASTALS.		Output to FAS.
SRC	Standard Requirements Code. Identifies type unit (TOE).	Input from FAS, FASTALS and TUCHA.	FSORT RSORT MATCH LOG	Pos 1-2 branch identity 2-5 type organization 6 series 7 year 8-9 variation 10 level 11-12 paragraph.
STACO	Station code. Army location code for unit home station. See ARLOC.	Input from FAS file.		Manually set for notional units.

Data element	Description	Set	Used	Remarks
STAGR	Structured aggregate strength. Full TOE strength.	Input from FAS. Input from FASTALS for use in generation of notional units.	LOG NUR MATCH	Output to FAS.
STENL	Structured enlisted strength.	Input from FAS. Input from FASTALS for use in generation of notional units.	MATCH	Output to FAS.
STNNM	See data element LOCNA. Location name (abbreviated) of unit's home station.	Input from FAS.	GEO	
STOFF	Structured officer strength.	Input from FAS. Input from FASTALS for generation of notional units.	MATCH	Output to FAS.
STWOF	Structured warrant officer strength.	Input from FAS. Input from FASTALS for generation of notional units.	MATCH	Output to FAS.
Theater Code	One character code designating the theater to which a unit is deploying.	LAYIN	GEO NUR	

Data element	Description	Set	Used	Remarks
Time Period	Time increment established in FASTALS.	Input from FASTALS.	RSORT MATCH	Output to FAS. RDD are based on Time Period as determined for a specific scenario.
TLAC	Type location or activity code.	Input from ARLOC.	SORTARLOC	Example: AIN = Army installation. See App B, DA Cir 525-10-4.
TMODE	Preferred mode of travel.	LOG NUR		Output to MORSA and SMOBSMOD. MORSA SMOBSMOD Air A 1 Sea S 2 Optional P 0 Mixed - 3
TPSN	Troop program sequence number. Code which groups units by type and size.	Input from FAS. Input from FASTALS for generation of notional units.		Output to FAS. See AR 18-19 for explanation.
TYPCO	Type code.	Input from FAS file.	FSORT	1 = TOE unit 2 = TDA augmentation 3 - TDA unit.
UGRID	Universal transmercator grid coordinates (military grid coordinates).	Input from ARLOC.	N/A	

Data element	Description	Set	Used	Remarks
UIC	Unit identification code. Code which uniquely identifies each unit.	Input from FAS. Input from POMCUS, generated by MATCH for notional units.	LOG	Output to FAS.
UIN	Unit identity number. Associates SRC with LRE within FASTALS.	Input from FASTALS	RSORT	Output to FAS.
ULCCC	See Level.			
UNMBR	Unit number. The numerical portion of the unit designation.	Input from FAS generated by MATCH for notional units.		Output to FAS.
UNTDS	Unit description.	Input from FAS and FASTALS.		Output to MORSA.

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

CHAPTER V
JOB PREPARATION

1. FMA Job Preparation

a. General. To run an FMA job, the user must obtain two basic input files, FAS and CAMPSUM, and prepare eleven manual input tables. The files and tables are described below.

b. Input Files

(1) FAS file. The FAS file is fully described in Reference 4. The file contains one or more records for each unit in the force. Multiple records occur when a unit changes status (e.g. receives an updated SRC), in which case the effective date of the change is shown on the applicable record. Each record is 307 characters. Fields applicable to CAMP have been described in Chapter IV and are shown in Table V-1. The ADCON and AUTHR fields are generally overlaid with other data. The ADCON field is used by the DA staff to pass selected data to CAA for match purposes. The ADC01 field contains the Theater Code which must be used in match processing as it identifies the theater to which a unit is eligible to deploy. The ADC02 and ADC03 fields may be used by the DA staff to identify a specific Time Period and RDD: in general, however, these values will not be specified by DA but are set during match processing. Match results are forwarded to DA in the AUTHR field.

(2) CAMPSUM. The FASTALS Requirements file is provided in CAMPSUM format as described in Table V-2. The file contains one record for each type unit requirement, except that each above-the-line unit is represented by its Headquarters Company. The FASTALS system is described in Reference 5.

c. Preliminary Processing. The FAS file is loaded into UDS using program BUILD. The user should analyze the force and make any required updates with UDS; then a UDS extract is performed to obtain the desired force for matching purposes. Program SORTUDS is then used to create the mass storage FAS file used by FMA. Once this file is obtained, any number of matches can be made against it.

d. Input Tables. FMA programs require the input tables described below. Normally, a different set of tables is developed by the user for each match.

TABLE V-1, FAS File Record

Position	Data element
2	FICOD
3	COMPO
4-9	UIC
10-15	EDATE
16	ACTCO
27-28	ASGMT
29-43	AUTHR
44-45	Branch
46-47	CARSS
58-62	DAMPL
71-72	DSCMP
80-82	FPLAN
83-87	JCSTY
88-90	LOCCO
96-98	MBLOC
101-109	MBSTA
119-124	ADCON
127	PHASE
130-132	ROBCO
134-145	SRC
147-151	STACO
154-162	STNNM
180-184	TPSN
186	TYPCO
194-197	UNMBR
198-203	PECOD
204-224	UNTDS
278-282	STOFF
283-287	STWOF
288-292	STENL
293-297	STAGR

TABLE V-2, CAMPSUM File Record

Position	Type	Description
1-3	N	FASTALS UIN
6-14	AN	SRC position 1-9
15-16	N	SRC paragraph
17	N	Personnel level
18	N	Equipment level
20-21	A	Branch
23-54	A	Unit description
56-58	N	FASTALS strength
60-61	N	FASTALS Time Period
64-67	N	FASTALS Rules Manual play Theater structure variable Existence rule Workload
69	N	FASTALS Logical Region of Employment
71	N	Combination rule a/
73	N	Rounding rule a/
75-79	N	TPSN
81-85	N	Officer strength b/
87-91	N	Warrant Officer strength b/
99-103	N	Enlisted strength b/
105-109	N	Total strength b/

a/ Not used in CAMP
b/ Used as structured strength for notional units

(1) RSORT input. Two card types are input to RSORT to cause various records to be deleted from the CAMPSUM file. Normally these records will correspond to units which are played by hand in FSORT. Card type 1 is described in Table V-3. Exactly one of this card type is required. All records with Time Period less than the input value (First Time Period) will be deleted from the CAMPSUM file. For example, if First Time Period equals two, then all units with Time Period = 1 (usually corresponding to in-country units) will be deleted. Card type 2 is described in Table V-4. One card type 2 is entered for each UIN to be deleted. All records with a specified UIN and Time Period later than or equal to the specified UIN will be deleted.

(2) FSORT input. Six input tables are required for FSORT.

(a) The Hand Played UIC data as described in Table V-5 are used to change selected data fields on a FAS record. One card is entered for each unit to be changed. If RDD is not specified, the specified fields will be changed, and the unit will be processed according to the changed data; the unit will not be processed to the Hand Played Unit file (HPU). If a new RDD is specified, the unit will be hand played by FMA.

(b) The Lock Out ADC01 data described in Table V-6 are used to specify theaters not to be considered in this match. One card is entered for each theater to be locked out. Units with specified Theater Code are not match candidates in FMA and are not deployed in MRG.

(c) The Alternate Theater ADC01 data described in Table V-7 are used to designate alternate theaters. One card is entered for each Theater Code representing an alternate theater. Units with a specified Theater Code are not considered for matching in FMA but are deployed in MRG. This table is also input to Program ALT.

(d) MCODE 1 (POMCUS) ROBCO data. One card is entered for each POMCUS ROBCO as described in Table V-8. POMCUS units are hand played.

(e) The MCODE 0 (in-country) data require two card types as described in Table V-9. In-country units are hand played.

(f) The Lock Out COMPO data described in Table V-10 are used to delete entire sets of units from the force.

TABLE V-3, RSORT Card Type 1

Position	Format	Data entry
1-3	I3	First Time Period

TABLE V-4, RSORT Card Type 2 (Lock Out UIN)

Position	Format	Data entry
1-3	I3	FASTALS UIN
5-6	I2	Time Period

TABLE V-5, Hand Played UIC Data

Position	Type	Data entry
1-6	A	UIC--Unit Identification Code of Force record to be changed or hand played.
8-10	A	New ROBCO (blank if no change). Appears in final output.
12	A	New Theater Code (blank if no change). Appears in AUTHR field of final product.
14-15	N	Hand Played Time Period (if blank, program will calculate correct time period to correspond to RDD).
17-19	N	Hand Played RDD. If this field is entered the unit will be hand played; if this field is blank, the unit will be processed normally in accordance with other changes.

TABLE V-6, Lock Out ADC01 Data

Position	Type	Data entry
1	A	ADC01--Theater Code. Locked out in FMA and not deployed in MRG.

TABLE V-7, Alternate Theater ADC01 Data

Position	Type	Data entry
1	A	ADC01--code to select alternate theater support units (TPSN less than 20000)
3-5 a/	N	Earliest RDD to assign to units with this ADC01
7-9 a/	N	Last RDD to assign to units with this ADC01
11-13 a/	N	Increment between previous values to assign RDD
15-17 a/	A	AUTHR 15--Theater Code to be used by MRG

a/these fields are used by ALT but not by FSORT

TABLE V-8, MCODE 1 (POMCUS) ROBCO Data

Position	Type	Data entry
1-3	A	ROBCO field values to be considered POMCUS unit--the asterisk character (*) may be used to indicate a don't care condition for any position of the field. Only COMPO 1 units are considered.
5-7	N	RDD to be assigned above-the-line units with this ROBCO. May be blank in which case RDD will be assigned by TPSN as in Table V-14.
9-11	N	RDD to be assigned to support units with this ROBCO.

TABLE V-9, MCODE 0 (In-country) Data

Position	Type	Data entry
1-3 a/	N	M-day for primary theater. RDD for in-country units.
1	A	ADCO1 value to be considered in-country in the primary theater. One card for each ADCO1 value.

a/first card only

TABLE V-10, Lock Out Compo Data

Position	Type	Data entry
1	N	COMPO value to be locked out. Records with these COMPO values are deleted from the force file.

(3) MATCH input. Three input tables are required by MATCH.

(a) The Subperiod RDD data described in Table V-11 are used to designate RDD to be assigned within each Time Period. Higher priority units within a given Time Period are assigned the earlier RDD within that Time Period. This table is also used by ATL to assign a Time Period to each above-the-line unit.

(b) Tables V-12 and V-13 are used to create FAS records for notional units. Normally the values to be entered are specified by the DA staff.

(4) ATL input. ATL requires the input described in Table V-14. One card is entered for each TPSN. RDD are normally determined by the DA staff.

2. MRG Job Preparation

a. General. To run an MRG job, the user must obtain the five basic input files (ARLOC, TUCHA, POMCUS, MORSA and FAS) and provide card image input to six programs. The ARLOC file is obtained quarterly from US Army Management Systems Support Agency. The file is documented in References 6 and 7. The TUCHA file used is the Army input to the JCS TUCHA file used in the Joint Operations Planning System (JOPS). The JCS TUCHA file is documented in Reference 8. This file is obtained from the United States Army Command and Control Support Agency (USACCSA) quarterly. The POMCUS file is a nonstandard file provided to CAA on request by ODCSLOG. The MORSA file is obtained on request from CCTC through JCS-J4. See Reference 3 for documentation. Note that this file contains the MORSA data record, not the MORSA transaction card image which is produced by CAMP. The FAS file is produced by FMA Program LAYIN. The files and input data are described in subsequent paragraphs.

b. Preprocessors. MRG requires four preprocessor programs identified in Chapter III to process the input tapes. These programs are required only when the tapes to be processed are updated. The user is encouraged to coordinate with Joint and Strategic Forces Directorate to determine the status of the preprocessor output files. If the files are current, the preprocessor runs can be eliminated.

(1) ARLOCSORT. This preprocessor produces the GEO file from the ARLOC file. The layout of the ARLOC file is shown in Table V-15. Manual inputs are not required.

TABLE V-11, Subperiod RDD Data

Position	Data entry
1-4	RDD 1st subperiod
5-8	RDD 2nd subperiod
9-12	RDD 3rd subperiod
13-16	RDD 4th subperiod
17-20	RDD 5th subperiod
21-24	RDD 6th subperiod
25-28	last day of time period

One card per time period-10 cards required

TABLE V-12, Notional Unit Fields

Card	Position	Type	Data entry
1	1	A	FICOD
2	1-6	N	EDATE
3	1	A	ACTCO
4	1-2	A	ASGMT
5	1	N	AUTH 8
6	1-2	N	CARSS
7	1-2	A	DSCMP
8	1-3	A	FPLAN
9	1-3	AN	LOCCO
10	1-3	AN	MBLOC
11	1-6	A	MBSTA pos 1-6
12	1-3	A	MBSTA pos 7-9
13	1	A	PHASE
14	1-5	AN	STACO
15	1-6	A	STNNM pos 1-6
16	1-3	A	STNNM pos 7-9
17	1	N	TYPCO
18	1	N	COMPO
19	1-2	AN	UIC pos 1-2
20	1-2	AN	UIC pos 1-2 host nation support unit (not currently used)

TABLE V-13, Notional Unit PECOD Data

Position	Type	Data entry
1-2	N	SRC pos 1-2
5-10	AN	PECOD value to be assigned

TABLE V-14, Above-the-Line RDD Data

Position	Type	Data entry
1-5	N	TPSN
8	N	Theater Code to be assigned
11-13	N	RDD to be assigned

TABLE V-15, ARLOC File Record Description

Data name	Description	Start pos	No char	Type
AREA	Geographic area	1	3	A/N
ARLOC	Army location code	9	5	A/N
GELOC	JCS geographic location code	18	4	A
LOCNA	Abbreviated location name	27	9	A/N
LOCNM	Location name	40	17	A/N
TLAC	Type installation	65	3	A
FCTCD	State or foreign country	78	18	A/N
UGRID	Universal transverse mercator grid coordinates	98	11	A/N
LFLLC	Latitude/longitude	112	15	A/N
STAT	Status of record	129	1	A

(2) TUCHAROLL. This preprocessor produces the TUSUM file. The TUCHA file is input to this program. Of the four record types in TUCHA, only the ABF1 and the F2 record types are used. The layout of these records is given in Tables V-16 and V-17. Manual input is not required.

(3) POMCUSROLL. This preprocessor is used to produce the POMSUM file. The POMCUS file which is input to this program, is in a modified TUCHA transaction format. The layout of these records is given in Tables V-18 through V-20. The F3 detail record is not

TABLE V-16, TUCHA ABF1 Record Description

Position	Type	Description
1-5	A/N	Unit type code
9	N	Record type = 1
13-15	A	Unit level code
16	N	Deployment indicator code
17	A	Service code
18	A	Record security classification
19-33	A/N	Short type name
34	A/N	Invalid record indicator (T for TYPEA record)
35	A/N	Unit type status (C for cancelled)
36	A	F1 security classification
37-90	A/N	Type name
91-96	A/N	Originators UIC
97-102	N	Date of record creation
103-108	N	Date of last change
109-113	N	Authorized wartime personnel
114-118	N	Nonorganic passengers
119-121	N	Number or cargo categories
122-124	N	Count of F2 records
125	A/N	New F1 indicator
126-137	A/N	SRC
138	A	blank

TABLE V-17, TUCHA F2 Record Description

Position	Type	Description
1-5	A/N	Unit type code
6-7	A/N	Cargo category code
8	A	Containerization code
9	N	Record type = 2
13	A	F2 security classification
14	A/N	New F2 indicator
15	A	Heavy lift code
18-23	N	Square feet of cargo
24-35	N	Short tons of cargo (in tenths)
36-41	N	Measurement tons of cargo
42-47	N	Barrels of POL (in hundreds of barrels)
48-50	N	Number of F3 records required
51-53	N	Count of F3 records

TABLE V-18, POMCUS F1 Record

Position	Type	Description
7	N	Record type = 1
9-13	A/N	Unit type code
19-23	N	Nonorganic passengers
24-26	N	Number of F2 records
71-76	A/N	UIC of POMCUS unit
90-101	A/N	SRC
102-120	A/N	Not used

TABLE V-19, POMCUS F2 Record

Position	Type	Description
7	N	Record type = 2
9-13	A/N	Unit type code
14	A	POMCUS code A = TAT, B = Other exclusion, C = Prepo
34-35	A/N	Cargo category code
36	A	Containerization code
37-42	N	Square feet of cargo
45-50	N	Short tons of cargo (in tenths)
52-57	N	Measurement tons of cargo
59	A	Heavy lift code
60	N	Number of F3 records
71-76	N	UIC of POMCUS unit
90-101	A/N	SRC

TABLE V-20, POMCUS F3 Record

Position	Type	Description
7	N	Record type = 3
9-13	A/N	Unit type code
14	A	POMCUS code
19-20	A/N	Cargo category code
21	A	Containerization code
22-24	N	F3 identification number
25-38	A/N	Cargo description
39-42	N	Length
43-45	N	Width
46-48	N	Height
49-52	N	Square feet
53-55	N	Number of pieces

currently used but is documented here because other references are not available. Manual inputs are not required by POMSUM, however manual editing of the output may be required to ensure that the UIC identified on the Force file are present on the POMSUM file.

(4) MORSAROLL. The MORSAROLL preprocessor extracts other service movement requirements from the MORSA file. The layout of the MORSA file is shown in Table V-21. Manual input card formats are shown in Table V-22.

c. MRG Processors. Input to MRG programs is described in Tables V-23 through V-28. The Node Table is input to program GEO to assign SMOBSMOD nodes and default GELOC to each AREA used as an origin or destination. One entry as described in Table V-24 is required for each AREA. The SMOBSMOD node number will always be selected from this table. The default GELOC will be selected only if the actual LOCNA is not found on the GEO file.

d. Postprocessors. Program INLAY requires one card input as described in Table V-29, Program Build requires no manual input.

TABLE V-21, MORSA File Record Description

Data name	Description a/	Start pos	No char	Type
PLAN-ID	Identification code for plan	1	1	N
SERV	Service	2	1	A/N
SEQ	Sequence number of this record	3	4	N
RT	Record type	12	1	A/N
RD	Record description	13	1	A/N
REC-NAME	Record (unit or package description)	14	12	A/N
LEVEL		26	3	A/N
TMODE		30	1	A
AVAIL		32	3	N
RDD		35	3	N
NAME-O	Origin name	44	14	A/N
CODE-2-O	MORSA origin node	61	2	A/N
GEO-O	Origin GELOC	68	4	A
CODE-2-D	MORSA destination node	117	2	A/N
GEO-D	Destination GELOC	124	4	A
PAX		128	5	N
ACAMO		133	5	N
ASPLY		138	5	N
BULK		143	5	N
OVER-POL	(1) ST oversize cargo or, (2) thousand barrels POL (if RT=P)	148	5	N
OUTSZ		153	5	N
NAT		158	5	N
TOTAL-WT	Total cargo weight (less POL)	163	5	N
TPSN		168	5	N
SRC		173	11	A/N
FRN	Force requirements number	184	4	A/N
JCSTY		188	5	A/N
UIC		193	6	A/N
NSDA		217	5	N
Filler	N/A	222	57	A/N

a/ Where description is not given see Data Dictionary, this document.

TABLE V-22, MORSAROLL data

Card type 1 (one card for each origin code on MORSA file)

Col	Data name	Type	Description
1-2	CODE-2-O	A	MORSA origin node
4-6	ONODE	N	SMOBSMOD origin node

Card type 2 (one card for each destination code on MORSA file)

Col	Data name	Type	Description
1-2	CODE-2-D	A	MORSA destination node
4-6	DNODE	N	SMOBSMOD destination node

Card type 3 (one card only)

Col	Data name	Type	Description
1-2	DELTA-RDD	N	Enter number of days rollup desired for other services on RDD. For example, an entry of 1Ø will result in the following:

Actual RDD	RDD assigned by MORSAROLL
0-9	5
10-19	15
20-29	25
etc	

Col	Data name	Type	Description
2-4	DELTA-AVAIL	N	Enter number of days rollup desired for other services on AVAIL. For example, an entry of Ø5 will result in the following:

Actual AVAIL	AVAIL assigned by MORSAROLL
0-4	2
5-9	7
10-14	12
etc	

TABLE V-23, GEO Data

Card type	Number required	Position	Type	Description
1	1	1	N	Number of destination theaters (number of type 2 cards to follow--maximum 7)
2	1 per theater	1	N	Theater Code to select input force records
		3-5	N	Day available at origin for all except reserve above-the-line units.
		7-10	A	Destination GELOC
		12-14	N	Destination node number
		16-18	N	Day available at origin for reserve component above-the-line units
3	1	1-2	N	Number of exception cards (type 4) to follow (maximum of 20)
4	1 per exception	1-5	N	TPSN
		7-10	A	Destination GELOC
		12-14	N	Destination node number
		16-18	N	Day available
		20-22	N	Required Delivery Date
		24	A	TMODE (P-optional, A-air, S-sea, Z-not moved)
		26	N	Destination theater
		28-31	A	Origin GELOC
		33-35	N	Origin node number

TABLE V-24, Node Table

Position	Type	Description
1-3	AN	AREA (position 1 blank if numeric)
4-12	AN	LOCNA
13-16	A	GELOC
17-21	AN	ARLOC
22-24	N	SMOBSMOD node number

TABLE V-25, LOG Data

Card type	Number required	Position	Type	Description
1	1	1	N	Number of destination theaters
		3	N	Number of theater sets
		5-11	N	Theater code for each destination theater (maximum 7--blank if less than 7)
		13-19	N	Theater set for each destination theater
		27	N	Plan identification for each theater set
2	1	1-4	A	APOE GELOC-not used
		6-9	A	SPOE GELOC-not used
		11-16	N	Accompanying supply (dry and packaged POL) for non-POMCUS units lbs/man--decimal point required
		18-23	N	Accompanying ammunition for non-POMCUS units lbs/man--decimal point required
		25-30	N	Accompanying supply (dry) for POMCUS units lbs/man--decimal point required
		32-37	N	Accompanying ammunition for POMCUS units lbs/man--decimal point required
3	1	1-2	N	Number of SRC substitution cards (type 4) to follow (may be zero)
4	1 per substitution	1-11	AN	SRC less ALO (position 10) that appears in Force but not on TUCHA
		13-23	AN	TUCHA SRC that was substituted for Force SRC

TABLE V-26, UPMDFI Data

Card	Position	Type	Description
1	1-2	N	Number of SRC substitution cards to follow
2	1-11	AN	SRC less ALO (position 10) that appears in Force but not on TUCHA
	13-23	AN	TUCHA SRC to be substituted for Force SRC

TABLE V-27, NUR Data (continued on next page)

Card	Position	Type	Description
1	1	N	Number of destination theaters
	3	N	Number of theater sets
	5-11	N	Theater Code for each destination theater
	13-19	N	Theater set for each destination theater
	21-27	N	Plan identification for each theater set
2	1	N	Theater set number
	3-5	N	D-Day--day on which personnel casualties start
	7-0	N	RDAY--day on which theater resupply build up begins
	11-13	N	BLDUP--day on which build up objective is to be met
	15-17	N	C-Day--all units with RDD less than or equal to C-Day are considered in-country
	19-21	N	A-Day--all resupply packages with RDD greater than or equal to A-Day will be assigned TMODE=S (sea)
	23-25	N	Last day of resupply period 1
	27-29	N	Last day of resupply period 2
	31-33	N	Last day of resupply period 3
	35-37	N	Last day of resupply period 4
	39-41	N	Last day of resupply period 5
	43-45	N	Last day of resupply period 6
	47-49	N	Last day of resupply period 7
	50-51	N	Last day of resupply period 8 (last day of consumption)
	55-57	N	Level of supply (days) required at BLDUP--dry
	59-61	N	Days of ammo required at BLDUP
	63-65	N	Days of POL required at BLDUP
	67-69	N	Minimum level (days) of supply (dry) prior to RDAY
	71-73	N	Minimum days of ammo prior to RDAY day of resupply
	75-78	N	Minimum days of POL prior to RDAY day of resupply

TABLE V-27, NUR Data (continued on next page)

Card	Position	Type	Description
3	1	N	Theater set number
	3-11	N a/	Short tons of dry PWRS
	13-21	N a/	Short tons of ammo PWRS
	23-31	N a/	Bbls of POL PWRS
	33-36	N a/	Fraction of in-country strength needed for filler
	38-40	N	RDD for 1st filler increment
	42-44	N	RDD for 2nd filler increment
	46-48	N	RDD for 3rd filler increment
	50-53	N a/	Fraction of total fillers in 1st increment
	55-58	N a/	Fraction of total fillers in 2nd increment
	60-63	N a/	Fraction of total fillers in 3rd increment
	65-68	A	Destination GEO code
	70-72	N	Destination node number
4	1	N	Theater set number
	3-7	N a/	Replacements/1000/day 1st period
	9-13	N a/	Replacements/1000/day 2nd period
	15-19	N a/	Replacements/1000/day 3rd period
	21-25	N a/	Replacements/1000/day 4th period
	27-31	N a/	Replacements/1000/day 5th period
	33-37	N a/	Replacements/1000/day 6th period
	39-43	N a/	Replacements/1000/day 7th period
	45-49	N a/	Replacements/1000/day 8th period
5	1	N	Theater set number
	3-8	N a/	Dry accompanying supply for non-POMCUS units lbs/man
	10-15	N a/	Accompanying ammo for non-POMCUS units lbs/man
	17-22	N a/	Accompanying POL for non-POMCUS units lbs/man
	24-29	N a/	Dry accompany supply for POMCUS units lbs/man
	31-36	N a/	Accompanying ammo for POMCUS units lbs/man
	38-43	N a/	Accompanying POL for POMCUS units lbs/man

---

a/Decimal point required

TABLE V-27, NUR Data (concluded)

Card	Position	Type	Description
6	1	N	Theater set number
	2	N	Resupply type (1-dry, 2-ammo, 3-POL)
	4-9	N a/	lbs/man/day 1st period
	11-16	N a/	lbs/man/day 2nd period
	17-22	N a/	lbs/man/day 3rd period
	24-29	N a/	lbs/man/day 4th period
	31-36	N a/	lbs/man/day 5th period
	38-43	N a/	lbs/man/day 6th period
	45-51	N a/	lbs/man/day 7th period
	53-58	N a/	lbs/man/day 8th period
7 b/	1-4	A	Origin GELOC
	6-8	N	Origin node number
	10-13	N a/	Fraction of dry from this origin
	15-18	N a/	Fraction of ammo from this origin

---

a/Decimal point required
b/20 type 7 cards are required-use blank if necessary-last two specify replacement and POL origins

TABLE V-28, PDKAGE Data

Card	Position	Type	Description
1	1-2	N	Packaging interval on RDD
	4-5	N	Packaging interval on availability
2	1	N	Theater number
	3-5	N	RDD to begin requiring all units to theater by mixed mode
	7	N	Theater number
	9-11	N	RDD to start all mixed mode
	13	N	Theater number
	15-17	N	RDD to start all mixed mode
	19	N	Theater number
	21-23	N	RDD to start all mixed mode
	25	N	Theater number
	27-29	N	RDD
	31	N	Theater number
	33-35	N	RDD
	37	N	Theater number
	39-41	N	RDD

TABLE V-29, INLAY Data

Position	Data entry
1	FICOD for NUR records
3-8	EDATE for NUR records

COMPUTER ASSISTED MATCH PROGRAM
(CAMP)

CHAPTER VI
JOB UTILIZATION

1. Force Match Algorithm. This section describes the output from FMA. The ultimate output is the FAS tape overlaid with match results. However, intermediate reports are produced that should be examined. Examples are shown in Chapter VIII. To analyze the final output appropriately, it is necessary to load the FAS tape into the UDS system using Program BUILD. Detailed and summary reports can then be obtained and analyzed.

a. Program SORTUDS. This program provides a listing of the force in COMPO/UIC sequence. Data elements displayed are identified in Table VI-1.

b. Program FSORT provides a copy of the manual entry data and a listing of each of the five files produced. Each listing has the same format as described in Table VI-2. The Match Candidate file is sorted in match sequence; all other files are in the original COMPO/UIC sequence.

c. Program RSORT displays the manual entry data, a count of the units, and two reports: the UIN report (deleted units) and the REQ report (requirements) in match sequence. The reports are described in Table VI-3.

d. Program ATL prints the manual entry tables, the input data file (ATL) and the output data file (MATL). The output file is exactly the same as the input file (Table VI-2) except that the ADCO2 and ADCO3 fields have been overlaid with Time Period and RDD, respectively.

e. Program ALT prints the manual entry table and the output file (MALT) that is exactly the same as the input file (Table VI-2) except that the ADCON field has been overlaid with Destination Theater and RDD.

f. Program MATCH produces a report (Table VI-4) that readily identifies matches, overages, and shortfalls. The first two data items displayed are the REQ-SRC (FASTALS requirement SRC) and FORCE-SRC (FAS file SRC). Entries in both columns indicate a match. An entry in FORCE-SRC and a blank REQ-SRC indicates an overage. An

TABLE VI-1, SORTUDS Report

Column	Data element
1-6	UIC
9-13	TPSN
16	COMPO
19-21	FPLAN
24-26	ROBCO
29-40	SRC
43-47	JCS Type Unit Code
50-51	Level
55-56	CARRS
29-62	Unit number
65-66	Branch
69-86	Unit description
99-107	Home station
109-111	LOCCO
113-114	DSCMP
117	ADCO1
119-120	ADCO2
122-124	ADCO3

TABLE VI-2, FSORT Reports

Column	Data element
1-7	SRC (positions 1-5, 8,9)
8	COMPO
9	MCODE
11-15	DAMPL
16-21	UIC
22-26	SRC (positions 6,7, 10-12)
27-28	Branch
29-31	Level
32-49	Unit description
50-54	TPSN
55-57	ROBCO
58-66	Home station
67-69	LOCCO
70-74	STAGR
75-80	ADCON
81-86	Sequence number
87-91	FSORT Key a/

a/FSORT Key has the following meaning:

Lock Out file: Reason for lock out
ATL file: YES or NO to indicate ADC03 set based on manual input to FSORT
Other files: blank

TABLE VI-3, RSORT Report

Column	Data element
1-5	SRC (truncated)
7-8	SRC (positions 8,9)
11	Time period
13-14	SRC (positions 6,7)
16	SRC (position 10)
18	SRC (position 12)
20	SRC (position 11)
22-24	Level
26-46	Unit description
47-50	Rules
52	Logical Region of Employment
54	Combining rule
56	Rounding rule
58-62	TPSN a/
64-68	STOFF a/
70-74	STWOF a/
76-80	STENL a/
82-86	STAGR a/
88-90	FASTALS strength
92-93	Branch
95-97	UIN

a/Used only in the generation of notional units.

TABLE VI-4, MATCH Report

Column	Data element
1-8	REQ-SRC
11-18	FORCE-SRC
23-24	Branch
28-30	Level
33-50	Unit description
54	COMPO
58-63	UIC
65-69	TPSN
71-73	UIN
76	Time Period
79-82	Requirment sequence number
84	MCODE
88-90	ROBCO
93-101	Home station
104-106	LOCCO
109-113	FASTALS strength
115-119	FAS strength[a]
121-125	Force sequence number

[a] For notional units, this is the structured aggregated strength from the TOE file.

entry in REQ-SRC and a blank in FORCE-SRC indicates a shortfall for which a notional unit has been generated. This report is prepared in SRC sequence to facilitate analysis. The Match Report is the key report of the match process as it identifies the required force structure (except for hand played units) and the status of the actual force in relation to the requirements.

g. Programs LAYIN and BINCOPY do not produce any significant printed output. However, the FAS tape produced by BINCOPY can be loaded into a keyed access UDS file using program BUILD. UDS provides an enhanced capability for analysis of the match results. Further, the file produced by LAYIN (FOUT) is used as input to MRG.

2. Movement Requirements Generator. In this section the output from MRG is described. The ultimate output, will be movement requirements in SMOBSMOD and/or MORSA format. However, intermediate output is available at several points in the processing to assist the user in analysis. Examples of output are shown in Chapter VIII.

a. Preprocessors. As stated in Chapter III, MRG includes four preliminary processors. Each one produces an intermediate file which is used in subsequent processing and a printed report that should be perused by the user. The use of these four reports is described in the following paragraphs.

(1) TUCHAROLL. The output from TUCHAROLL is a summary of movement characteristics of each type unit. The report is printed in SRC sequence and includes two lines per type unit. The data elements are described in Table VI-5. Under each cargo category, the first line displays weights in tenths of short tons, and the second line displays square feet of cargo if appropriate. Occasionally, the user may discover errors in the TUCHA file. The correct data can usually be ascertained through coordination with ODCSLOG, and the intermediate mass storage file produced by TUCHAROLL, in the same format as the printed report, can be corrected using the edit processor. The output from TUCHAROLL is also useful in determining appropriate SRC substitutions. If the force being processed contains SRC not on the TUCHA file, substitutions must be made subsequent to running program LOG.

(2) POMCUSROLL. The output from POMCUSROLL is similar to that from TUCHAROLL except that the report is sorted in UIC sequence and four lines are printed for each unit. The first two lines indicate quantities to be moved and correspond to the first two lines in the TUCHAROLL report. The next two lines indicate quantities prepositioned. The POMCUS report should be checked against the force to ascertain if all POMCUS units are represented. If not, substitutions can be made using the edit processor, or a new POMCUS file can be obtained from ODCSLOG.

TABLE VI-5, Data Elements Displayed by TUCHAROLL

Data element		Description
SRC		Standard requirements code, line number
ULC/UTC	line 1: line 2:	Unit level code/unit type code Unit description
PAX	line 1: line 2:	Number passengers Unit strength
A0		Vehicles, non-air transportable (NAT)
B0		Non-self-deployable aircraft (NSDA) NAT
B1		NSDA outsize
A1		Vehicles, outsize
J1		Nonvehicular outsize cargo
B2C		NSDA oversize, containerizable
A2C		Vehicles, oversize, containerizable
A2D		Vehicles, oversize, noncontainerizable
J2D		Nonvehicular cargo, oversize, noncontainerizable
J2C		Nonvehicular cargo, oversize, containerizable
A3C		Vehicles, Bulk, containerizable
J3C		Nonvehicular cargo, bulk, containerizable
J7C		Accompanying dry bulk supplies
AMMO		Accompanying ammunition
UE/B	line 1: line 2:	Total weight of unit equipment Total square feet of NSDA
AS/A	line 1: line 2:	Total accompanying supply weight[a] Total square feet of vehicles

a/Not used in MRG

(3) MORSAROLL. The reports are produced by MORSAROLL. The first is a listing of the SMOBSOD input data (A and B cards only, the C card is produced but not listed) for other service movements. The A, B, and C cards are described in Reference 2. The report is listed in package number sequence as derived by sorting the other service movement data on RDD, availability date, travel mode, origin node, and destination node. The format is exactly the same as that for SMOBSMOD input. The A card shows package number, RDD, and availability date. One B card is produced for each cargo category included in the package. Current categories arelisted at Table VI-6. Each B card displays availability date, destination node, origin node, travel mode, number of passengers (thousands) cargo quantity thousands of barrels of POL or thousands of short tons for other cargo) and the cargo type. The second report is a detailed listing of the units making up each package. The data elements on this report are self-explanatory except for OV/POL. Normally all cargo quantities are input from MORSA in short tons, and the OV/POL field displays quantity of oversize cargo. However, if the cargo is POL (record type 'P') then the OV/POL field displays barrels in thousands. This program also provides a listing of all records for which the origin or destination could not be found in the node tables. A code (1 for origin, 9 for destination) indicates which type node was not located. Also displayed are the MORSA node identification, total weight, bulk, number of passengers, RDD, and availability date. Although the program does not error terminate when a node is not found, the user should update the node tables and assign a SMOBSMOD node to each MORSA node.

(4) ARLOCSORT. Two reports are provided by ARLOCSORT. The first report displays the entire Army Location file in Army Location code (ARLOC) sequence and in abbreviated location name (LOCNA) sequence. The data elements displayed have been listed in Table V-15. The second report is sorted by AREA and displays only AREA, LOCNA, GELOC and ARLOC. These are the same data elements contained in the GEO Location file and subsequently used by Program GEO.

b. Main processors. Each of the MRG programs provides output to assist the user in analysis of results.

(1) GEO. All manual entry data are formatted and displayed to the user. The user should carefully check the program output to verify the accuracy of manual entry data. Also displayed is the number of records in the GEO file and Node table. The GEO file is the same file output by ARLOCSORT, so the number of records should agree. The Node table is created from card input, so the number of

records can easily be checked. The last direct output from GEO identifies the number of locations on the Force file that could not be found on the GEO file and the average number of accesses to the GEO file per unit. Locations not found on the GEO file are assigned a notional GELOC from the Node table based on LOCCO or MBLOC. If the LOCCO and MBLOC values are not found in the Node table, the program will terminate abnormally after identifying all such values. The user should then update the Node table. GEO also produces a Movement Data file 1 (MDF1) which can be listed or edited separately by the user. This file displays the data elements shown in Table VI-7.

(2) LOG. This program may terminate abnormally if there are discrepancies in the input data. The user must correct the discrepancies and rerun the program in the event of abnormal termination of the program. The next paragraphs describe the output under abnormal and normal conditions.

(a) Abnormal termination. If any SRC appearing on the MDF1 file are not found on the TUSUM file, these SRC will be listed. The user must then determine appropriate substitute SRC whose movement characteristics will be used in lieu of the SRC that could not be found. Programs UPMDF1 and LOG are then rerun with the list of substitute SRC as input.

(b) Normal termination. All manual entry data are formated and displayed to the user who should carefully check this output to verify the accuracy of the manual entry data. The number of units in each theater set is displayed. A list of units, that are identified as POMCUS units on the Force file but that are not located on the POMCUS file, is then printed. Such mismatches do not cause program termination because these units will be assigned zero deployment weight. However, the user should peruse the list to determine if an error exists on one of the files. Program LOG also produces a file of MORSA cards for each unit. This file can be printed or edited as desired by the user.

(3) UPMDF1. This program does not produce any user output.

(4) NUR. Manual entry data are formated and displayed for user perusal. This output should be carefully checked to assure correctness. NUR prints a supply status summary in 5-day increments for each theater. Data elements are listed in Table VI-8. NUR also identifies units on MDF1 for which no corresponding Theater Code was input to the program. No resupply or replacement quantities are generated for these units. If the user wants these quantities determined, appropriate manual entry data must be provided to NUR. NUR also produces a file of MORSA cards for the nonunit movement requirements. This file can be listed or edited by the user as desired.

TABLE VI-6, SMOBSMOD Cargo Categories

Category	Cargo type
1	PAX
2	VEH-NAT
3	NSDA-NAT
4	NSDA-OUT
5	VEH-OUT
6	POL
7	NV-OUT
8	NSDA-OVER-C
9	NSDA-OVER-NC
10	VEH-OVER-C
11	VEH-OVER-NC
12	NV-OVER-NC
13	NV-OVER-C
14	VEH-BULK-C
15	NV-BULK-C
16	DRY RESUPPLY
17	AMMO

TABLE VI-7, MDF1 Data Elements

Data element	Description
SRC	Standard requirement code
UIC	Unit identification number
TPSN	Troop program sequence number
ORIGIN	Origin GELOC
Destination	Destination GELOC
Origin-node	SMOBSMOD node number
Destination-node	SMOBSMOD node number
RDD	Required delivery date
AVAIL	Availability date
MODE	MORSA travel mode
Theater code	See Data Dictionary (this document)
MCODE	See Data Dictionary (this document)
Unit description	Description of the type unit

TABLE VI-8, Supply Status Summary

Data element	Description
DAY	Effective day of summary
THR	Theater code
UNITS	Number of units deployed to the theater (cumulative)
STRENGTH	Authorized theater strength used to determine number of casualties
REPLACEMENTS	Number of replacements required
TYPE	'1' = Dry bulk cargo '2' = Ammunition '3' = POL
DAYS OH	Number of days of supply on hand in theater stocks (by type)
DAYS DESRD	Number of days of supply required to be in theater stocks (by type)
TONS OH	Number of short tons on hand
TONS REQRD	Shortfall in theater stocks (POL units and thousands of barrels)

(5) MERGEMORSA. A user output is not produced by this program.

(6) PCKAGE and PRTPCK. The printed output is self-explanatory. Two files are also produced that can be listed or edited by the user. One file contains SMOBSMOD input (A, B, and C cards) as defined in paragraph 2a(3) of this chapter. The second file contains a detail listing of units (identified by UIC and TPSN) within each Army package.

c. Post processors. MRG includes two programs designed to interface with UDS. Use of these programs is optional with the user but is strongly recommended because of the versatile report generaion capability of UDS. The UDS reports provide a valuable aid to the user in analysis of MRG results. Two programs are required to produce a UDS file. Program INLAY overlays movement data on selected fields of the FAS file, creates additional FAS records for nonunit movements, and produces a modified FAS tape. Program BUILDMRG accepts the modified FAS tape and produces a UDS file with the new data. Table VI-9 identifies the MRG data items overlaid on the FAS file and the UDS data names associated with the overlaid data. A special version of the UDS report generator (39FACT.MAINABS) should be used to extract reports from this file. See Reference 1 for an explanation of UDS.

TABLE VI-9, MRG Data Items Overlaid on FAS File

MRG data	Size	FAS position	UDS name
Availability Date	3	34-36	AVAL1
Record Type	1	37	RTCCC
Record Description	1	38	RDCCC
Sequence Number	4	48-53	CCNUM
Origin Geographic Location Code	4	54-57	ORGEO
Destination Geographic Location Code	4	58-61	DEGEO
Travel Mode	1	62	TMODE
Required Delivery Date	3	110-112	REQDA
Destination node a/	3	113-115	DNODE
Orgin node a/	3	116-118	ONODE
Package Number a/	4	163-167	PKGNR
Unit level b/	3	204-206	UNTDS
Unit description b/	12	207-224	UNTDS
Bulk (short tons)	6	228-232	BULK1
Oversize (short tons)	6	233-237	OVRSZ
Outsize (short tons)	6	238-242	OUTSZ
NAT (short tons)	6	243-247	NATAA
NSDA (square feet)	6	248-252	NSDA1
Total weight (short tons)	6	253-257	DEPST
Accompanying ammunition (short tons)	6	258-262	ACAMO
POL (000 barrels)	6	263-267	POLBB
Passengers	6	268-272	PAXAA
Accompanying dry cargo (short tons)	6	273-277	ASPLY

a/Data items obtained from SMOBSMOD input. All other data items obtained from MORSA input.

b/Overlaid for nonunit records only.

APPENDIX A

STUDY CONTRIBUTORS

PRECEDING PAGE BLANK-NOT FILMED

COMPUTER ASSISTED MATCH PROCESS (CAMP)

APPENDIX A
STUDY CONTRIBUTORS

1. Study Director

Mr Gary L. Martin, Joint and Strategic Forces Directorate

2. Study Team

Ms Patricia Fleming, Methodology and Resources Directorate
Mr Ivan Keller, Force Concepts and Design Directorate
MAJ Ernest R. Montagne, Jr, Joint and Strategic Forces Directorate
Mr Rene Plourde, Methodology and Resources Directorate
Ms Avonale Stephenson, Project Planning and Control Office

3. Support Personnel

Ms Betty Spence
Ms Armin Marvin
Ms Vivian Layman

APPENDIX B

REFERENCES

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

APPENDIX B
REFERENCES

1. US Army Concepts Analysis Agency, "Capabilities and Use of the CAA Unit Data System (UDS)," Draft Documentation, Bethesda, MD, 23 Jul 74.

2. __________, "Strategic Mobility Simulation Model (SMOBSMOD) User's Manual," Draft Documentation, Bethesda, MD, 30 May 74.

3. Defense Communications Agency, "Movement Requirements for Staff Planning and Special Studies Applications (MORSA)," User's Manual, Logistic Data Branch (ATTN: C318), Command and Control Technical Center. To be published.

4. Dept of Army, "Force Accounting System (FAS)," User's Guide, Washington, DC, Oct 72.

5. Research Analysis Corporation, "The FOREWON Force-Planning System," App C of "User's Guide for Force Analysis Simulation of Theater Administrative and Logistic Support (FASTALS)," RAC-R-86, May 71.

6. Dept of Army, "Army Location Codes, States Within the United States," DA Circular 525-10-3, Washington, DC, 30 Aug 74.

7. __________, "Army Location Codes, Foreign Locations," DA Circular 525-10-4, Washington, DC, 30 Aug 74.

8. Defense Communications Agency, "Type Unit Data File (TUCHA)," in Users Manual of Computer System Manual, CSM UM 157-74, Jul 74.

9. Dept of Army, "Troop Program Sequence Number," AR 18-19, Washington, DC, 16 Aug 72.

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

APPENDIX C
GLOSSARY

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

APPENDIX C
Glossary

Term	Definition
ALT	A computer program. A file output by program FSORT for input to program ALT
ammo	ammunition
ATL	Above-the-line. Refers to major combat forces (divisions and separate brigades). All units with TPSN first position 0 or 1 are above-the-line. A computer program. A computer file output by program FSORT for input to program ATL
BINCOPY	A computer program
BUILD	A computer program
BLDUP	day stockage objective is to be achieved
BUILDMRG	A computer program
CAA	Concepts Analysis Agency
CAMP	Computer Assisted Match Program
CAMPSUM	A program RSORT input computer file
CAN	A program Match data input file. A program FSORT output file. The match candidate file
CCTC	Command and Control Technical Center
COBOL	common business oriented language
DA	department of the Army
DC	display and compute
DO	display only

Term	Definition
FAS	Force Accounting System
FASTALS	Force Analysis Simulation of Theater Administration and Logistic Support
FMA	force match algorithm
FORTRAN	Formula Translation (a scientific programing language for computers)
FSORT	A computer program
GEO	A computer program that produces the Movement Data File 1
HPU	Hand played units. A computer file output by program FSORT for input to program LAYIN
INLAY	A computer program
JCS	Joint Chiefs of Staff
K	thousand
LAYIN	A computer program
LOG	A computer program
LOU	Locked out units. Units not considered for matching. A computer file output by program FSORT for input to program LAYIN
MALT	A program LAYIN computer input file. Also a program ALT output file
MATCH	A computer program
MATL	A file output by program ATL for input to program LAYIN
MERGEMORSA	A computer program
MORSA	Mobility Requirements for Staff Analysis
MORSAROLL	A computer program that provides preliminary processing for data base building

Term	Definition
MRD	Methodology and Resources Directorate, CAA
MREQ	A computer file output by program MATCH for input to program LAYIN (includes matches, overages, and notional units)
NUR	A computer program
ODCSOPS	Office, Deputy Chief of Staff for Operations and Plans
PCKAGE	A computer program
POL	petroleum, oils and lubricants
POMCUS	prepositioning of materiel configured to unit sets. A computer file input to MRG
POMCUSROLL	A computer program that provides preliminary processing for data base building
POMSUM	A summary file produced by POMCUSROLL
pos	position
PRTPCK	A computer program
PWRS	prepositioned war reserve stock
RDAY	day theater resupply build up begins
RDD	required delivery date
REQ	A computer file output by program RSORT for input to program MATCH
ROLLUP-MODIFIED	A new version of CAMPSUM
RSORT	A computer program
SORTARLOC	A computer program
SORTUDS	A computer program that provides preliminary processing for data base building
TOE	table(s) of organization and equipment

Term	Definition
TOE/TPSN	table(s) of organization and equipment/Troop Program Sequence Number. A computer file
TPSN	Troop Program Sequence Number (see reference 9)
TUCHA	Type unit data, a computer file input to MRG
TUCHAROLL	A computer program that provides preliminary processing for data base building
TUSUM	A summary file produced by TUCHAROLL
UDS	Unit Data System. Also, a computer file
UPMDF1	A computer program
WWMCCS	worldwide military command and control system
66 ROLLOUT	A third version of CAMPSUM (used if movement requirements are to be generated)
66 ROLLUP	The principal version of Program CAMPSUM

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

CHAPTER VII

PROGRAM LISTINGS

BEST AVAILABLE COPY

```
@ELT,L   24MONTAGNE.BUILD
ELT007 RLIB70 08/16-07:40:07-(3,)
     000   010010 IDENTIFICATION DIVISION.
     000   010020 PROGRAM-ID. 'BUILD'.
     000   010030 AUTHOR. J MEYEROWITZ.
     000   010040 DATE WRITTEN. 02 MAY 1973.
     000   010060 REMARKS. READ A FAS FILE FROM TAPE
     000   010070                  307 CHARACTERS BLOCKED 24 PER RECORD.
     000   010080            WRITE A UDF FILE TO MASS STORAGE USING ISFM
     000   010090                  360 CHARACTERS PER RECORD.
     000   020010 ENVIRONMENT DIVISION.
     000   020020 CONFIGURATION SECTION.
     000   020030 SOURCE-COMPUTER. UNIVAC-1108.
     000   020040 OBJECT-COMPUTER. UNIVAC-1108.
     000   020050 INPUT-OUTPUT SECTION.
     000   020060 FILE-CONTROL.
     000   020070     SELECT MASTER-FILE ASSIGN TO MASS-STORAGE FORCE
     000   020080     ACCESS MODE IS RANDOM SYMBOLIC KEY IS REAL-KEY
     000   020085     ORGANIZATION IS INDEXED FILE-DESCRIPTION IS FILE-DESCRIPT.
     000   020120     SELECT FAS-FILE ASSIGN TO UNISERVO FAS-FILE.
     000   020130     SELECT PRINT-FILE ASSIGN TO PRINTER.
     000   030010 DATA DIVISION.
     000   030020 FILE SECTION.
     000   030030 FD  MASTER-FILE
     000   030040     LABEL RECORDS ARE STANDARD
     000   030050     DATA RECORD IS MASTER-REC.
     000   030060 01  MASTER-REC            PICTURE IS X(10752).
     000   031010 FD  FAS-FILE
     000   031020     LABEL RECORD IS OMITTED
     000   031030     RECORD CONTAINS 7368 CHARACTERS
     000   031035     RECORDING MODE IS 1
     000   031040     DATA RECORD IS TAPE-BLOCK.
     000   031050 01  TAPE-BLOCK            PICTURE IS X(7368).
     000   032010 FD  PRINT-FILE
     000   032020     RECORD CONTAINS 132 CHARACTERS
     000   032030     LABEL RECORD IS OMITTED
     000   032040     DATA RECORD IS PRINT-LINE.
     000   032050 01  PRINT-LINE            PICTURE IS X(132).
     000   040010 WORKING-STORAGE SECTION.
     000   040020 77  NDEX                  PICTURE IS 99 COMP SYNC RIGHT VALUE IS 0.
     000   040030 77  N                     PICTURE IS H9       VALUE IS ZERO.
     000   040100 01  WORK-BLOCK.
     000   040110     02  WORK-REC  OCCURS 24 TIMES.
     000   040120         03  PART-1.
     000   040130             04  FILLER PICTURE IS X.
     000   040140             04  WORKID.
     000   040143               05 FICOD PICTURE IS X.
     000   040144               05 COMPO PICTURE IS X.
     000   040145               05 UICCC PICTURE IS X(6).
     000   040146               05 EDATE PICTURE IS X(6).
     000   040150             04  FILLER PICTURE IS X(11).
     000   040160         03  PART-2.
     000   04162              04  FILLER     PIC X(12).
     000                  04  THEATER-CODE     PIC X.
     001                  04  FILLER           PIC X(31).
     001                  04  DSCMP     PIC XX.
     001                  04  FILLER       PIC XXX.
```

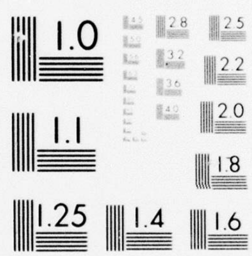
```
000   04164            04  ESCON      PIC X.
000   04166            04  FILLER     PIC X(108).
000                 04  TRCON             PIC X.
003                 04  FILLER            PIC X(122).
000   050010 01  DATA-REC.
000   050020     02  SEG-1            PICTURE IS X(26).
000   050030     02  FILLER           PICTURE IS X          VALUE IS 'X'.
000   050040     02  SEG-2            PICTURE IS X(333).
000   060010 01  PRT-AREA.
000   060020     02  PRT-ITEM         PICTURE IS X(18)   OCCURS 7 TIMES.
000   080010 01  ER-LINE.
000   080020     02  FILLER         PICTURE IS X(12)   VALUE IS 'INVALID KEY '.
000   080030     02  ER-KEY         PICTURE IS X(14).
000   150010 01  REAL-KEY.
000   150020     02  REC-KEY          PICTURE IS X(14).
000   150030     02  FILLER           PICTURE IS X(4).
000   180010 01  FILE-DESCRIPT.
000   180020     02  NUMBER-OF-RECORDS          PICTURE IS H9(10) VALUE IS 12000.
000   180030     02  RECORD-LENGTH              PICTURE IS H9(10) VALUE IS 360.
000   180040     02  RECORD-LENGTH-MAX          PICTURE IS H9(10) VALUE IS 0.
000   180050     02  RECORD-KEY-LENGTH          PICTURE IS H9(10) VALUE IS 18.
000   180060     02  OVERFLOW-AREA              PICTURE IS H9(10) VALUE IS 10.
000   180070     02  I-O-F                      PICTURE IS H9(10) VALUE IS 1000.
000   180080     02  DATA-NAME-8.
000   180090         03  DATA-NAME-9   PICTURE IS X(3).
000   180100         03  DATA-NAME-10  PICTURE IS X(3).
000   190010 01  INFORM.
000   190020     02  NUMBER-OF-BLOCKS PICTURE IS H9(10).
000   190030     02  NUMBER-OF-INDX-BLOCKS  PICTURE IS H9(10).
000   190040     02  NUMBER-OF-OVERFL-BLOCKS  PICTURE IS H9(10).
000   190050     02  NUMBER-OF-RECORDS          PICTURE IS H9(10).
000   190060     02  NUMBER-OF-RECORDS-IN-IOF  PICTURE IS H9(10).
000   190070     02  NUMBER-OF-RECORDS-DELRTD  PICTURE IS H9(10).
000   190080     02  NUMBER-OF-RECORDS-READ  PICTURE IS H9(10).
000   190090     02  NUMBER-OF-RECORDS-READ-FROM-IF    PICTURE IS H9(10).
000   190100     02  NUMBER-OF-RECORDS-WRITTEN         PICTURE IS H9(10).
000   195010 01  INFORM-FL-DATA.
000   195020     02  NUMBER-OF-BLOCKS PICTURE IS 9(6).
000   195030     02  NUMBER-OF-INDX-BLOCKS  PICTURE IS 9(6).
000   195040     02  NUMBER-OF-OVERFL-BLOCKS  PICTURE IS 9(6).
000   195050     02  NUMBER-OF-RECORDS          PICTURE IS 9(6).
000   195060     02  NUMBER-OF-RECORDS-IN-IOF  PICTURE IS 9(6).
000   195070     02  NUMBER-OF-RECORDS-DELRTD  PICTURE IS 9(6).
000   195080     02  NUMBER-OF-RECORDS-READ  PICTURE IS 9(6).
000   195090     02  NUMBER-OF-RECORDS-READ-FROM-IF    PICTURE IS 9(6).
000   195100     02  NUMBER-OF-RECORDS-WRITTEN         PICTURE IS 9(6).
000   200000 PROCEDURE DIVISION.
000   200010 START-TEST.
000   200020     OPEN INPUT FAS-FILE WITH NO REWIND.
000   200030     OPEN OUTPUT PRINT-FILE.
000   200040     OPEN OUTPUT MASTER-FILE.
000   210000 TAPE-READ.
000   210010     MOVE ZERO TO NDEX.
000   210020     READ FAS-FILE AT END GO TO FINI.
000   210030     MOVE TAPE-BLOCK TO WORK-BLOCK.
000   220000 GET-ITEM.
000   220010     ADD 1 TO NDEX.
```

```
000  220020        IF NDEX = 25 GO TO TAPE-READ.
000  220021         IF WORKID (NDEX) = SPACES GO TO GET-ITEM.
000  220025        IF WORKID (NDEX) = 'bbbbbbbbbbbbbb' GO TO GET-ITEM.
000                MOVE THEATER-CODE(NDEX) TO TRCON(NDEX).
000  220030        MOVE PART-1 (NDEX) TO SEG-1.
000  220040        MOVE PART-2 (NDEX) TO SEG-2.
000  220050        MOVE WORKID (NDEX) TO REAL-KEY.
000  220055        ADD 1 TO N.
000  220060        IF N = 8 PERFORM LINE-LIST THRU X-LIST.
000  220065        WRITE MASTER-REC FROM DATA-REC INVALID KEY GO TO ERR-0.
000  220070        MOVE REAL-KEY TO PRT-ITEM (N).
000  220080        GO TO GET-ITEM.
000  225010 LINE-LIST.
000  225020        MOVE 1 TO N.
000  225030        WRITE PRINT-LINE FROM PRT-AREA.
000  225040        MOVE SPACES TO PRT-AREA.
000  225050 X-LIST.
000  225060        EXIT.
000  226010 ERR-0.
000  226020        MOVE REAL-KEY TO ER-KEY.
000  226025        WRITE PRINT-LINE FROM ER-LINE.
000  226030        GO TO FINI.
000  230000 FINI.
000  230010        CLOSE FAS-FILE.
000  230020        CLOSE MASTER-FILE USING INFORM.
000  230030        MOVE CORRESPONDING INFORM TO INFORM-FL-DATA.
000  230035        WRITE PRINT-LINE FROM PRT-AREA.
000  230040        WRITE PRINT-LINE FROM INFORM-FL-DATA AFTER 2 LINES.
000  230050        CLOSE PRINT-FILE.
000  230060        STOP RUN.

END ELT.

ƏHDG    FMA/MRG PRE-PROCESSOR PROGRAM SORTUDS ••UNCLASSIFIED•• •L,0
```

```
@ELT,L    32MARTIN.SORTUDS2FAS
ELT007 RLIB70 08/16-07:40:09-(1,)
000001      000         IDENTIFICATION DIVISION.
000002      001         PROGRAM-ID.                 SORTUDS2FAS
000003      000         ENVIRONMENT DIVISION.
000004      000         CONFIGURATION SECTION.
000005      000         SOURCE-COMPUTER.            UNIVAC-1108.
000006      000         OBJECT-COMPUTER.            UNIVAC-1108.
000007      000         INPUT-OUTPUT SECTION.
000008      000         FILE-CONTROL.
000009      000             SELECT PRNTFL ASSIGN TO PRINTER.
000010      000             SELECT CARDFL ASSIGN TO CARD-READER.
000011      000             SELECT UDSFL ASSIGN TO UNISERVO SRC-TAPE.
000012      000             SELECT FASFL ASSIGN TO UNISERVO UIC-TAPE.
000013      000             SELECT SRTFL ASSIGN TO MASS-STORAGE XA.
000014      000         DATA DIVISION.
000015      000         FILE SECTION.
000016      000         FD  PRNTFL
000017      000             LABEL RECORD OMITTED
000018      000             DATA RECORD PRNTLINE.
000019      000         01  PRNTLINE.
000020      000             02  NUMB                PIC 9(5).
000021      000             02  FILLER              PIC X(2).
000022      000             02  MESSAGE             PIC X(125).
000023      000         FD  CARDFL
000024      000             LABEL RECORD OMITTED
000025      000             DATA RECORD CARD.
000026      000         01  CARD.
000027      000             02  TPSN                PIC 9(5).
000028      000             02  FILLER              PIC X(1).
000029      000             02  ADCON               PIC 9(6).
000030      000             02  FILLER              PIC X(68).
000031      000         FD  UDSFL
000032      000             BLOCK CONTAINS 24 RECORDS
000033      000             RECORD CONTAINS 360  CHARACTERS
000034      000             LABEL RECORD STANDARD
000035      000             DATA RECORD IS UDS.
000036      000         01  UDS.
000037      000                 03  RECDEL          PIC X.
000038      000                 03  FICOD           PIC X.
000039      000                 03  COMPO           PIC X.
000040      000                 03  UICCC.
000041      000                     04  UIC12       PIC X(2).
000042      000                     04  UIC36       PIC X(4).
000043      000                 03  EDATE           PIC 9(6).
000044      000                 03  ACTCO           PIC A.
000045      000                 03  AMSCO           PIC X(11).
000046      000                 03  ASGMT           PIC X(2).
000047      000                 03  AUTHR           PIC X(15).
000048      000                 03  BRNCH           PIC A(2).
000049      000                 03  CARSS           PIC X(2).
000050      000                 03  CCNUM           PIC X(6).
000051      000                 03  CHGNR           PIC X(2).
000052      000                 03  CIVCN           PIC X(2).
000053      000                 03  DAMPL           PIC X(5).
000054      000                 03  DEPLO           PIC X(3).
000055      000                 03  UNPID           PIC X(2).
```

AD-A048 554 ARMY CONCEPTS ANALYSIS AGENCY BETHESDA MD F/G 15/7
COMPUTER ASSISTED MATCH PROGRAM (CAMP),(U)
AUG 76 G L MARTIN, E R MONTAGNE
UNCLASSIFIED CAA-D-76-5 NL

2 OF 4
AD
A048554



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
@ELT,L   32MARTIN.SORTUDS2FAS
ELT007 RLIB70 08/16-07:40:09-(1,)
000        IDENTIFICATION DIVISION.
001        PROGRAM-ID.                     SORTUDS2FAS
000        ENVIRONMENT DIVISION.
000        CONFIGURATION SECTION.
000        SOURCE-COMPUTER.                UNIVAC-1108.
000        OBJECT-COMPUTER.                UNIVAC-1108.
000        INPUT-OUTPUT SECTION.
000        FILE-CONTROL.
000            SELECT PRNTFL ASSIGN TO PRINTER.
000            SELECT CARDFL ASSIGN TO CARD-READER.
000            SELECT UDSFL ASSIGN TO UNISERVO SRC-TAPE.
000            SELECT FASFL ASSIGN TO UNISERVO UIC-TAPE.
000            SELECT SRTFL ASSIGN TO MASS-STORAGE XA.
000        DATA DIVISION.
000        FILE SECTION.
000        FD  PRNTFL
000            LABEL RECORD OMITTED
000            DATA RECORD PRNTLINE.
000        01  PRNTLINE.
000            02  NUMB                    PIC 9(5).
000            02  FILLER                  PIC X(2).
000            02  MESSAGE                 PIC X(125).
000        FD  CARDFL
000            LABEL RECORD OMITTED
000            DATA RECORD CARD.
000        01  CARD.
000            02  TPSN                    PIC 9(5).
000            02  FILLER                  PIC X(1).
000            02  ADCON                   PIC 9(6).
000            02  FILLER                  PIC X(68).
000        FD  UDSFL
000            BLOCK CONTAINS 24 RECORDS
000            RECORD CONTAINS 360  CHARACTERS
000            LABEL RECORD STANDARD
000            DATA RECORD IS UDS.
000        01  UDS.
000                03  RECDEL              PIC X.
000                03  FICOD               PIC X.
000                03  COMPO               PIC X.
000                03  UICCC.
000                    04  UIC12           PIC X(2).
000                    04  UIC36           PIC X(4).
000                03  EDATE               PIC 9(6).
000                03  ACTCO               PIC A.
000                03  AMSCO               PIC X(11).
```

```
000056   000        03  MBPRD        PIC X(3).
000057   000        03  DSCMP        PIC X(2).
000058   000        03  ELSEQ        PIC X(2).
000059   000        03  EQCON        PIC X.
000060   000        03  ESCON        PIC X.
000061   000        03  ELAGG        PIC X.
000062   000        03  FNCAT        PIC X.
000063   000        03  FORCO        PIC X.
000064   000        03  FPLAN.
000065   000            04  FPLAN1   PIC X.
000066   000            04  FPLAN2   PIC X.
000067   000            04  FPLAN3   PIC X.
000068   000        03  JCSTY        PIC X(5).
000069   000        03  LOCCO        PIC X(3).
000070   000        03  MACTO        PIC X(3).
000071   000        03  MBCMD        PIC X(2).
000072   000        03  MBLOC        PIC X(3).
000073   000        03  FILLR        PIC X(2).
000074   000        03  MBSTA        PIC X(9).
000075   000        03  MILCN        PIC X(2).
000076   000        03  NOTID        PIC X.
000077   000        03  NTREF        PIC X(2).
000078   000        03  OESTS        PIC X.
000079   000        03  OPAGY        PIC 9(3).
000080   000        03  ADCON.
000081   000            04 ADCO1     PIC 9.
000082   000            04 ADCO2     PIC 99.
000083   000            04 ADCO3     PIC 999.
000084   000        03  OPDAT        PIC X.
000085   000        03  CATCO        PIC X.
000086   000        03  PHASE        PIC X.
000087   000        03  PRCON        PIC X.
000088   000        03  REPCO        PIC X.
000089   000        03  ROBCO.
000090   000            04  ROBCO1   PIC X.
000091   000            04  ROBCO2   PIC X.
000092   000            04  ROBCO3   PIC X.
000093   000        03  SPLIT        PIC A.
000094   000        03  SRCTO        PIC X(13).
000095   000        03  STACO        PIC X(5).
000096   000        03  STATS        PIC X(2).
000097   000        03  STNNM        PIC X(9).
000098   000        03  TDATE        PIC 9(5).
000099   000        03  TDNUM        PIC X(11).
000100   000        03  TMCCC        PIC A.
000101   000        03  TPSNA        PIC 9(5).
000102   000        03  TRCON        PIC X.
000103   000        03  TYPCO        PIC 9.
000104   000        03  UNCAP        PIC X.
000105   000        03  UNCON        PIC X.
000106   000        03  UNCLC        PIC X(5).
000107   000        03  UNMBR        PIC X(4).
000108   000        03  PECOD        PIC X(6).
000109   000        03  ULCCC        PIC A(3).
000110   000        03  UNTDS        PIC X(18).
000111   000        03  VCHNR        PIC X(3).
000112   000        03  AUSTR        PIC X(25).
```

```
000              03  OPSTR          PIC X(25).
000              03  STSTR          PIC X(25).
000              03  DPMNT          PIC X(4).
000              03  ENDRC          PIC X.
000              03  FILLER         PIC X(52).
000      FD  FASFL
             RECORD CONTAINS 7368 CHARACTERS
000          LABEL RECORD OMITTED
000          RECORDING MODE IS 1
000          DATA RECORD IS FAS-BLK.
000      01  FAS-BLK.
000          02  FAS-REC OCCURS 24 TIMES.
000              03  FAS-DATA       PIC X(307).
000      SD  SRTFL
000          DATA RECORD IS SRT.
000      01  SRT.
000              03  RECDEL         PIC X.
000              03  FICOD          PIC X.
000              03  COMPO          PIC X.
000              03  UICCC.
000                  04  UIC12      PIC X(2).
000                  04  UIC36      PIC X(4).
000              03  EDATE          PIC 9(6).
000              03  ACTCO          PIC A.
000              03  AMSCO          PIC X(10).
000              03  ASGMT          PIC X(2).
000              03  AUTHR.
000                  04  UIN        PIC X(3).
000                  04  TP         PIC 9(2).
000                  04  RDD        PIC 9(3).
000                  04  ESC        PIC X.
000                  04  EQC        PIC X.
000                  04  THR        PIC 9.
000                  04  FLG        PIC X(4).
000              03  BRNCH          PIC A(2).
000              03  CARSS          PIC X(2).
000              03  CCNUM          PIC X(6).
000              03  CHGNR          PIC X(2).
000              03  CIVCN          PIC X(2).
000              03  DAMPL          PIC X(5).
000              03  DEPLO          PIC X(3).
000              03  UNPID          PIC X(2).
000              03  MBPRD          PIC X(3).
000              03  DSCMP          PIC X(2).
000              03  ELSEQ          PIC X(2).
000              03  EQCON          PIC X.
000              03  ESCON          PIC X.
000              03  ELAGG          PIC X.
000              03  FNCAT          PIC X.
000              03  FORCO          PIC X.
000              03  FPLAN.
000                  04  FPLAN1     PIC X.
000                  04  FPLAN2     PIC X.
000                  04  FPLAN3     PIC X.
000              03  JCSTY          PIC X(5).
000              03  LOCCO          PIC X(3).
000              03  MACTO          PIC X(3).
```

```
000                  03  MBCMD           PIC X(2).
000                  03  MBLOC           PIC X(3).
000                  03  FILLR           PIC X(2).
000                  03  MBSTA           PIC X(9).
000                  03  MILCN           PIC X(2).
000                  03  NOTID           PIC X.
000                  03  NTREF           PIC X(2).
000                  03  OESTS           PIC X.
000                  03  OPAGY           PIC 9(3).
000                  03  ADCON.
000                      04 ADCO1        PIC 9.
000                      04 ADCO2        PIC 99.
000                      04 ADCO3        PIC 999.
000                  03  OPDAT           PIC X.
000                  03  CATCO           PIC X.
000                  03  PHASE           PIC X.
000                  03  PRCON           PIC X.
000                  03  REPCO           PIC X.
000                  03  ROBCO.
000                      04  ROBCO1      PIC X.
000                      04  ROBCO2      PIC X.
000                      04  ROBCO3      PIC X.
000                  03  SPLIT           PIC A.
000                  03  SRCTO           PIC X(13).
000                  03  STACO           PIC X(5).
000                  03  STATS           PIC X(2).
000                  03  STNNM           PIC X(9).
000                  03  TDATE           PIC 9(5).
000                  03  TDNUM           PIC X(11).
000                  03  TMCCC           PIC A.
000                  03  TPSNA           PIC 9(5).
000                  03  TRCON           PIC X.
000                  03  TYPCO           PIC 9.
000                  03  UNCAP           PIC X.
000                  03  UNCON           PIC X.
000                  03  UNCLC           PIC X(5).
000                  03  UNMBR           PIC X(4).
000                  03  PECOD           PIC X(6).
000                  03  ULCCC           PIC A(3).
000                  03  UNTDS           PIC X(18).
000                  03  VCHNR           PIC X(3).
000                  03  AUSTR           PIC X(25).
000                  03  OPSTR           PIC X(25).
000                  03  STSTR           PIC X(25).
000                  03  DPMNT           PIC X(4).
000                  03  ENDRC           PIC X.
000          WORKING-STORAGE SECTION.
000          77  BRC                     PIC 9(5).
000          01  WORK-SRC.
000              02  WSRC1-5.
000              03  WSRC1-3                PIC X(3).
000              03  WSRC4-5                PIC X(2).
000              02  WSRC612.
000              03  WSRC6-9                PIC X(4).
000              03  WSRCALO                PIC X(1).
000              03  WSRC11-12              PIC X(2).
000          01  FAS.
```

```
000          03  RECDEL        PIC X.
000          03  FICOD         PIC X.
000          03  COMPO         PIC X.
000          03  UICCC.
000              04  UIC12     PIC X(2).
000              04  UIC36     PIC X(4).
000          03  EDATE         PIC 9(6).
000          03  ACTCO         PIC A.
000          03  AMSCO         PIC X(10).
000          03  ASGMT         PIC X(2).
000          03  AUTHR.
000              04  UIN       PIC X(3).
000              04  FLG       PIC X(4).
000              04  MATCH     PIC X.
000              04  TP        PIC 9(2).
000              04  RDD       PIC 9(3).
000              04  LRE       PIC X.
000              04  THR       PIC 9.
000          03  BRNCH         PIC A(2).
000          03  CARSS         PIC X(2).
000          03  CCNUM         PIC X(6).
000          03  CHGNR         PIC X(2).
000          03  CIVCN         PIC X(2).
000          03  DAMPL         PIC X(5).
000          03  DEPLO         PIC X(3).
000          03  UNPID         PIC X(2).
000          03  MBPRD         PIC X(3).
000          03  DSCMP         PIC X(2).
000          03  ELSEQ         PIC X(2).
000          03  EQCON         PIC X.
000          03  ESCON         PIC X.
000          03  ELAGG         PIC X.
000          03  FNCAT         PIC X.
000          03  FORCO         PIC X.
000          03  FPLAN.
000              04  FPLAN1    PIC X.
000              04  FPLAN2    PIC X.
000              04  FPLAN3    PIC X.
000          03  JCSTY         PIC X(5).
000          03  LOCCO         PIC X(3).
000          03  MACTO         PIC X(3).
000          03  MBCMD         PIC X(2).
000          03  MBLOC         PIC X(3).
000          03  FILLR         PIC X(2).
000          03  MBSTA         PIC X(9).
000          03  MILCN         PIC X(2).
000          03  NOTID         PIC X.
000          03  NTREF         PIC X(2).
000          03  OESTS         PIC X.
000          03  OPAGY         PIC 9(3).
000          03  ADCON.
000              04 ADCO1      PIC 9.
000              04 ADCO2      PIC 99.
000              04 ADCO3      PIC 999.
000          03  OPDAT         PIC X.
000          03  CATCO         PIC X.
000          03  PHASE         PIC X.
```

```
000            03  PRCON          PIC X.
000            03  REPCO          PIC X.
000            03  ROBCO.
000                04  ROBCO1     PIC X.
000                04  ROBCO2     PIC X.
000                04  ROBCO3     PIC X.
000            03  SPLIT          PIC A.
000            03  SRCTO          PIC X(13).
000            03  STACO          PIC X(5).
000            03  STATS          PIC X(2).
000            03  STNNM          PIC X(9).
000            03  TDATE          PIC 9(5).
000            03  TDNUM          PIC X(11).
000            03  TMCCC          PIC A.
000            03  TPSNA          PIC 9(5).
000            03  TRCON          PIC X.
000            03  TYPCO          PIC 9.
000            03  UNCAP          PIC X.
000            03  UNCON          PIC X.
000            03  UNCLC          PIC X(5).
000            03  UNMBR          PIC X(4).
000            03  PECDD          PIC X(6).
000            03  ULCCC          PIC A(3).
000            03  UNTDS          PIC X(18).
000            03  VCHNR          PIC X(3).
000            03  AUSTR          PIC X(25).
000            03  OPSTR          PIC X(25).
000            03  STSTR          PIC X(25).
000            03  DPMNT          PIC X(4).
000            03  ENDRC          PIC X.
000        01  PRNTWORK.
000            03  UICCC.
000                04  UIC12      PIC X(2).
000                04  UIC36      PIC X(4).
000            03  FILLER         PIC XX  VALUE SPACES.
000            03  TPSNA          PIC 9(5).
000            03  FILLER         PIC XX  VALUE SPACES.
000            03  COMPO          PIC X.
000            03  FILLER         PIC XX  VALUE SPACES.
000            03  FPLAN.
000                04  FPLAN1     PIC X.
000                04  FPLAN2     PIC X.
000                04  FPLAN3     PIC X.
000            03  FILLER         PIC XX  VALUE SPACES.
000            03  ROBCO.
000                04  ROBCO1     PIC X.
000                04  ROBCO2     PIC X.
000                04  ROBCO3     PIC X.
000            03  FILLER         PIC XX  VALUE SPACES.
000            03  SRCTO          PIC X(12).
000            03  FILLER         PIC XX  VALUE SPACES.
000            03  JCSTY          PIC X(5).
000            03  FILLER         PIC XX  VALUE SPACES.
000            03  ULCCC          PIC X(3).
000            03  FILLER         PIC XX  VALUE SPACES.
000            03  CARSS          PIC X(2).
000            03  FILLER         PIC XX  VALUE SPACES.
```

```
000              03  UNMBR           PIC X(4).
000              03  FILLER          PIC XX  VALUE SPACES.
000              03  BRNCH           PIC X(2).
000              03  FILLER          PIC XX  VALUE SPACES.
000              03  UNTDS           PIC X(15).
000              03  FILLER          PIC X   VALUE SPACE.
000              03  MBSTA           PIC X(9).
000              03  FILLER          PIC X   VALUE SPACE.
000              03  MBLOC           PIC X(3).
000              03  FILLER          PIC X   VALUE SPACE.
000              03  STNNM           PIC X(9).
000              03  FILLER          PIC X   VALUE SPACE.
000              03  LOCCO           PIC X(3).
000              03  FILLER          PIC X   VALUE SPACE.
000              03  DSCMP           PIC X(2).
000              03  FILLER          PIC XX  VALUE SPACES.
000              03  ADCON.
000                  04  ADCO1       PIC 9.
000                  04  FILLER      PIC X   VALUE SPACES.
000                  04  ADCO2       PIC 99.
000                  04  FILLER      PIC X   VALUE SPACES.
000                  04  ADCO3       PIC 999.
000          PROCEDURE DIVISION.
000          OPENERS.
000              OPEN INPUT UDSFL CARDFL  OUTPUT FASFL PRNTFL.
000              READ CARDFL AT END MOVE 99999 TO TPSN IN CARD.
000              MOVE ZERO TO BRC.
000          SORT-IT.
001              SORT SRTFL ON ASCENDING KEY COMPO UICCC
000                INPUT PROCEDURE IS SORT-IN THRU UDS-EOF
000               OUTPUT PROCEDURE IS SORT-OUT.
000               GO TO CLOSEOUT.
000          SORT-IN.
000               READ UDSFL AT END GO TO UDS-EOF.
000               MOVE CORRESPONDING UDS TO FAS.
000               RELEASE SRT FROM FAS.
000               GO TO SORT-IN.
000          UDS-EOF.
000               EXIT.
000          SORT-OUT.
000               RETURN SRTFL INTO FAS AT END GO TO LAST-BLK.
000               IF TPSN IN CARD EQUALS TPSNA IN FAS
000                   MOVE ADCON IN CARD TO ADCON IN FAS.
000               MOVE CORRESPONDING FAS TO PRNTWORK.
000               WRITE PRNTLINE FROM PRNTWORK.
000               ENTER FORTRAN FCOPY SUBROUTINE REFERENCING FAS.
000               ADD 1 BRC.
000               MOVE FAS TO FAS-REC (BRC).
000               IF BRC EQUALS 24  WRITE FAS-BLK  MOVE ZERO TO BRC.
000               GO TO SORT-OUT.
000          LAST-BLK.
000               IF BRC EQUALS ZERO GO TO EOF-FAS.
000               ADD 1 TO BRC.
000               MOVE ALL '9' TO FAS-DATA (BRC).
000               IF BRC LESS THAN 24  GO TO LAST-BLK.
000               WRITE FAS-BLK.
000           EOF-FAS.
```

```
000                 ENTER FORTRAN ECOPY SUBROUTINE REFERENCING FAS.
000             CLOSEOUT.
000                 CLOSE UDSFL FASFL PRNTFL CARDFL.
000                 STOP RUN.

END ELT.


@ELT.L   32MARTIN.FCOPY
ELT007 RLIB70 08/16-07:40:13-(0,)
000             SUBROUTINE FCOPY(FAS)
000             DIMENSION FAS(52)
000             DATA IL/0/
000      C
000             WRITE(2)FAS
000             IL=IL+1
000             IF(IL/500*500.EQ.IL)PRINT 1,IL
000           1 FORMAT(I10)
000             RETURN
000      C
000      C
000      C     PUT  EOF ON BINARY FILE
000             ENTRY ECOPY(FAS)
000             PRINT 99,IL
000          99 FORMAT()
000             ENDFILE 2
000             RETURN
000      C
000      C
000      C
000      C        READ BINARY FILE
000             ENTRY FREAD(FAS)
000             READ(2,END=999) FAS
000             RETURN
000     999     FAS(1)=6H999999
000             FAS(2)=6H999999
000             REWIND 2
000             RETURN
000             END

END ELT.


@HDG    FMA PROGRAM FSORT                    ••UNCLASSIFIED•• •L,0
```

```
@ELT,L   F5FS.CKALT
ELT007 RLIB70 08/16-07:40:15-(3,)
003          SUBROUTINE CKALT (ADCO1,SIZE,ALTTAB,ALTFLG)
000          IMPLICIT INTEGER (A-Z)
000          INTEGER  ALTTAB(100)
000          LOGICAL  ALTFLG
002     C    WRITE (6,1) ADCO1,SIZE,(ALTTAB(K),K=1,SIZE)
002        1 FORMAT (' SUB CKALT  ADCO1=',A1,', SIZE=',I4, 1X, 5A6)
000     C
000          ALTFLG = .FALSE.
000          I = 1
000     C
000     C  DOWHILE  (NOT END OF TABLE ALTTAB)
000     C
000     1000 IF (I .GT. SIZE)  RETURN
000     C
000          IF (ADCO1 .NE. ALTTAB(I))  GOTO 1100
000     C      THEN
000                ALTFLG = .TRUE.
000                RETURN
000     C      ELSE
000     1100       I = I + 1
000     C    END-IF
000          GOTO 1000
000     C  END-DO
000          END

END ELT.


@ELT,L   F5FS.CKHPU
ELT007 RLIB70 08/16-07:40:16-(1,)
000     C                 C K H P U   -   R O U T I N E
000     C
000     C                                                     RENE' R. PLOURDE
000     C                                                     USACAA/MRD  JUN 1976
000     C
000     C    THIS ROUTINE CHECKS THE CURRENT FASREC VIA THE INFORMATION CONTAINED IN
000     C  HPINFO AGAINST THE INFORMATION CONTAINED IN THE HAND-PLAYED-UNIT TABLE +PUTA+
000     C  TO DETERMINE IF THE FASREC IS TO BE HAND-PLAYED.
000     C    A GIVEN FASREC IS HAND-PLAYED IF ITS UIC (HPINFO(1)) EQUALS A HAND-PLAYED
000     C  UIC FROM THE HAND-PLAYED TABLE (HPUTAB(I,1)), WHERE 0 < I < LIMIT +1.
000     C
000     C    GIVEN THAT THE FASREC IS A HAND-PLAYED-UNIT,(UIC'S MATCH) THEN ROBCO, ADCO1
000     C  ADCO2, AND ADCO3  ARE SET EQUAL TO THE CORRESPONDING FIELD FROM HPUTAB IF THE
000     C  FIELD IN HPUTAB IS NON-BLANK.  ALSO IF ADCO3 FIELD IS CHANGED THEN RDDFLG IS
000     C  SET TO .TRUE. TO INDICATE THAT THE RDD IS BEING HAND-PLAYED FOR THIS UNIT
000     C
000          SUBROUTINE  CKHPU (UIC,ROBCO,ADCO1,ADCO2,ADCO3,LIMIT,
000          *                  HPUTAB,RDDFLG)
000          IMPLICIT  INTEGER  (A - Z)
000          INTEGER   HPUTAB(100,6)
000          DATA   BLANK/6H      /
000          LOGICAL  RDDFLG
000     C
000     C     WRITE(6,1) UIC,ROBCO,ADCO1,ADCO2,ADCO3,LIMIT
```

```
000       1  FORMAT(' SUB CHPU  UIC=',A6,', ROBCO=',A3,',ADCO1=',A1,
000          * ', ADCO2=',A2,', ADCO3=',A3,', LIMIT=',I4)
000    C
000    C     HPUTAB(I,1) =   UIC,  HPUTAB(I,2) = ROBCO,  HPUTAB(I,3) = ADCO1,
000    C     HPUTAB(I,4) = ADCO2,  HPUTAB(I,5) = ADCO3,  HPUTAB(I,6) = FLAG
000    C
000    C  FIRST SET  RDDFLG TO FALSE THEN CHECK TO SEE IF HPUTAB IS EMPTY
000    C
000           RDDFLG  =  .FALSE.
000           IF  (LIMIT .EQ. 0)   RETURN
000           I = 1
000    C
000    C  DOWHILE   (I < LIMIT)
000    C
000     1000 IF  (I .GT. LIMIT)  RETURN
000    C
000    C      COMPARE  UIC'S   (FASREC UIC (HPINFO(I))  AND  HAND-PLAYED (HPUTAB(I,1))
000    C
001    C      UICFLG  =  CKUIC (UIC,HPUTAB(1,1))
001           IF  (UIC .EQ. HPUTAB(I,1))  GOTO 1100
000    C         THEN
000    C                UIC'S DID NOT MATCH THEREFORE INCREMENT POINTER AND CHECK NEXT ONE
000    C
000                     I = I + 1
000                     GOTO  1900
000    C         ELSE
000    C                UIC'S MATCH THEREFORE CHECK ROBCO AND ADCON FIELS FROM HPUTAB
000    C                AND CHANGE HPINFO FIELDS WHEN HPUTAB FIELDS ARE NON-BLANK
000    C                CHECK  HAND-PLAYED ROBCO FIELD FOR NON-BLANK
000     1100                IF  (HPUTAB(I,2) .EQ. BLANK)  GOTO 1200
000    C                   THEN
000    C                        CHANGE  FASREC  ROBCO  TO HAND-PLAYED ROBCO
000                             ROBCO  =  HPUTAB(I,2)
000    C                   ELSE
000    C                        HAND-PLAYED ROBCO IS BLANK THUS FASREC ROBCO IS NOT CHGED
000    C                END-IF
000    C                CHECK HAND-PLAYED ADCO1 FIELD FOR NON-BLANK
000     1200            IF  (HPUTAB(I,3) .EQ. BLANK)  GOTO 1300
000    C                   THEN
000    C                        CHANGE FASREC ADCO1  TO  HAND-PLAYED ADCO1
000                             ADCO1  =  HPUTAB(I,3)
000    C                   ELSE
000    C                        HAND-PLAYED ADCO1 IS BLANK THUS FASREC ADCO1 IS NOT CHGED
000    C                END-IF
000    C                CHECK HAND-PLAYED ADCO2 FIELD FOR NON-BLANK
000     1300            IF  (HPUTAB(I,4) .EQ. BLANK)  GOTO 1400
000    C                  THEN
000    C                       CHANGE FASREC ADCO2  TO  HAND-PLAYED ADCO2
000                            ADCO2  =  HPUTAB(I,4)
000    C                  ELSE
000    C                       HAND-PLAYED ADCO2 IS BLANK THUS FASREC ADCO2 IS NOT CHANGED
000    C
000    C                END-IF
000    C                CHECK HAND-PLAYED ADCO3 FIELD FOR NON-BLANK
000     1400            IF  (HPUTAB(I,5) .EQ. BLANK)  GOTO 1500
000    C                   THEN
000    C                        CHANGE FASREC ADCO3 TO HAND-PLAYED ADCO3  AND SET RDDFLG
```

```
000                    ADCO3  =  HPUTAB(I,5)
000                    RDDFLG =  .TRUE.
000   C          ELSE
000   C                HAND-PLAYED ADCO3 IS BLANK THUS NO-CHG TO FASREC
000   C             END-IF
000   C             SET POINTER TO ONE GREATER THAN LIMIT SO THAT WE EXIT FROM DOWHILE
000   1500          HPUTAB(I,6)   =  1
000                 I  =  LIMIT + 1
000   C       END-IF
000   1900 CONTINUE
000           GOTO 1000
000   C   END-DO
000   C
000           END

END ELT.

@ELT,L   FSFS.CKLO
ELT007 RLIB70 08/16-07:40:17-(2,)
000   C                C K L O   -   R O U T I N E
000   C
000   C                                                    RENE' R. PLOURDE
000   C                                                    USACAA/MRD JUN 1976
000   C
000   C      THIS ROUTINE IS USED TO DETERMINE WHETHER OR NOT A GIVEN FAS REC IS TO BE
000   C    LOCKED OUT.  IF ANY ONE OF THE FOLLOWING  FIVE CONDITIONS IS TRUE THE LOCK-
000   C    OUT-FLAG (LOFLAG) IS SET TO  .TRUE.
000   C
000   C       1.  ADCO1 FROM FAS RECORD MATCHES AN ADCO1 FROM THE LOCK-OUT TABLE (LOTAB)
000   C           PROVIDED FOR BY USER
000   C
000   C       2.  ALL TDA UNITS  (SRC IS BLANK  OR  TYPCO = 2 OR 3)
000   C
000   C       3.  SPLIT UNITS  (UIC 1ST POS IS A W  AND  UIC 6TH POS IS > 0)
000   C
000   C       4.  DISPLAY ONLY UNITS
000   C
000   C       5.  HAND-PLAYED COMPO'S
000   C
000           SUBROUTINE  CKLO (ADCO1,SRC15,TYPCO,UIC,DSCMP,COMPO,ADCO2,ADCO3,
002         *                 SIZEL,SIZEC,LOTAB,HPCTAB,LOFLAG,WHY)
000   C
000           INTEGER  LOTAB(100), HPCTAB(100)
000           IMPLICIT INTEGER  (A - Z)
000           LOGICAL  LOFLAG
000   C
000           DATA  BLANK/1H /, TWO/1H2/, THREE/1H3/, W/1HW/, L/1HL/
000           DATA   DC/2HDC/,  ZERO/48/
000   C
000   C       WRITE (6,2) ADCO1,SRC15,TYPCO,UIC,DSCMP,COMPO,ADCO2,
000   C     * ADCO3,SIZEL,SIZEC
000         2  FORMAT(' SUB CKLO  ADCO1=',A1,', SRC15=',A5,
000         * ', TYPCO=',A1,', UIC=',A6,', DSCMP=',A2,', COMPO=',A1,
000         * ', ADCO2=',A2,', ADCO3=',A3,', SIZEL=',I4,', SIZEC=',I4)
000   C
```

```
000          LOFLAG  =  .FALSE.
000    C
000    C  ***  CHECK CONDITION 1  (ADC01 FROM LOTAB)  ***
000    C
000          I = 1
000    C
000    C  DOWHILE   (NOT EOF FOR LOTAB)
000    C
000     1000 IF  (I .GT. SIZEL)  GOTO 2000
000    C
000          IF  (ADC01 .NE. LOTAB(I))  GOTO 1100
000    C        THEN
000                   LOFLAG  =  .TRUE.
001                   WHY = 6HADC01
000                   RETURN
000    C        ELSE
000     1100          I  =  I + 1
000    C     END-IF
000          GOTO 1000
000    C  END-DO
000    C
000    C  ***  CHECK CONDITION 2  (TDA UNITS)  ***
000    C
002     2000 IF  (SRC15 .NE. 0 )  GOTO 2100
000    C        THEN
000                   LOFLAG  =  .TRUE.
001                   WHY = 6HNOSRC
000                   RETURN
000    C        ELSE
000     2100          IF  (TYPCO .NE. TWO)  GOTO 2200
000    C                   THEN
000                             LOFLAG  =  .TRUE.
001                              WHY = 6HTYPC02
000                             RETURN
000    C                   ELSE
000     2200                    IF  (TYPCO .NE. THREE)  GOTO 2300
000    C                             THEN
000                                      LOFLAG  =  .TRUE.
001                                      WHY = 6HTYPC03
000                                      RETURN
000    C                             ELSE
000    C                                 NO-OP
000    C                        END-IF
000     2300                    CONTINUE
000    C              END-IF
000     2400          CONTINUE
000    C     END-IF
000     2500 CONTINUE
000    C
000    C  ***  CHECK CONDITION 3  (SPLIT UNITS)  ***
000    C
000     3000 DECODE (6,1,UIC)  UIC1ST, UIC6TH
000        1 FORMAT (A1,4X,R1)
000    C
000          IF  (UIC1ST .NE. W)  GOTO 3100
000    C        THEN
000                   IF  (UIC6TH .LE. ZERO)  GOTO 3100
```

```
000  C                 THEN
000                          LOFLAG  =  .TRUE.
001                          WHY = 6HSPLIT
000                          RETURN
000  C                 END-IF
000  C      END-IF
000   3100 CONTINUE
000  C
000  C  •••  CHECK CONDITION 4  (DISPLAY ONLY UNITS  •••
000  C
000   4000 IF  (DSCMP .EQ. DC)  GOTO 4100
000  C          THEN
000                    LOFLAG  =  .TRUE.
001                    WHY = 6HDO
001                    RETURN
000  C      END-IF
000   4100 CONTINUE
000  C
000  C  ••• CHECK CONDITION 5  (HAND-PLAYED COMPO'S)  •••
000  C
000   5000 I = 1
000  C
000  C  DOWHILE  (NOT EOF FOR HPCTAB TABLE)
000  C
000   5100 IF  (I .GT. SIZEC)  GOTO 6000
000  C
000         IF  (COMPO .NE. HPCTAB(I))  GOTO 5200
000  C          THEN
000                    LOFLAG  =  .TRUE.
001                    WHY = 6HCOMPO
000                    RETURN
000  C          ELSE
000   5200             I  =  I + 1
000  C      END-IF
000   5300 CONTINUE
000         GOTO 5100
000  C
000  C  END-DO
000  C
000   6000 CONTINUE
000  C
000         RETURN
000         END
```

END ELT.

```
@ELT,L   FSFS.CKSRC
ELT007 RLIB70 08/16-07:40:19-(1,)
000  C                C K S R C   -   R O U T I N E
000  C
000  C                                                   RENE' R. PLOURDE
000  C                                                   USACAA/MRD JUNE 1976
000  C
000  C     THIS ROUTINE CHECKS SRC POSITIONS 8 AND 9 TO DETERMINE WHETHER OR NOT TO
000  C   CHANGE THEM TO 99 OR LEAVE UNCHANGED.
```

```
000     C
000     C
000               SUBROUTINE   CKSRC (SRC15,SRC89)
000     C
000               IMPLICIT  INTEGER  (A - Z)
000     C
000               DATA  A/1HA/, Z/1HZ/, N99/2H99/
000     C
000               DECODE (2,1,SRC89)   SRC8, SRC9
000             1 FORMAT(2A1)
000     C
000     C         WRITE (6,2) SRC15, SRC89
000     2         FORMAT(' SUB CKSRC    SRC15=',A5,', SRC89=',A2)
000     C
000     C
001               IF  (SRC15 .EQ. 55018)  GOTO 2000
001               IF  (SRC15 .EQ. 12067)  GOTO 2000
000     C
000               IF  (SRC8.GE.A .AND. SRC8.LE.Z)  GOTO 1000
000     C            THEN
000                          SRC89  =  N99
000                          GOTO 2000
000     C            ELSE
000        1000              IF  (SRC9.GE.A .AND. SRC9.LE.Z)  GOTO 2000
000     C                       THEN
000                                     SRC89  =  N99
000     C                       ELSE
000     C                               NO-CHANGE
000     C                    END-IF
000     C         END-IF
000        2000 CONTINUE
000               RETURN
000               END

END ELT.


@ELT,L   F5FS.CMCODE
ELT007 RL1B70 08/16-07:40:19-(5,)
001     C                C  M  C  O  D  E   -   R  O  U  T  I  N  E
001     C
001     C                                                               RENE' R. PLOURDE
001     C                                                               USACAA/MRD JUNE 1976
001     C
001     C        THIS ROUTINE COMPUTES THE MCODE, (1, 2, OR 3) BY USING THE TABLES MCOTAB
001     C     AND MCITAB, WHICH ARE PROVIDED FOR BY THE USER.
001     C
001     C
004               SUBROUTINE   CMCODE (TYPE,ADCO1,ROBCO,ADCO3,MCODE,COMPO,SIZE1,
005            *                          SIZEO,MCOTAB,MCITAB,RDDFLG)
001     C
001                IMPLICIT INTEGER  (A-Z)
001     C         WRITE (6,99) TYPE,ADCO1,ROBCO,ADCO3,MCODE,SIZE1,SIZEO
001        99     FORMAT(1X,'TYPE=',I1,'ADCO1=',A1,'ROBCO=',A3,'ADCO3=',
001            *  A3,'MCODE=',A1,'SIZE1=',I3,'SIZEO=',I3)
001               INTEGER  TYPE, MCOTAB(100), MCITAB(100,3)
```

```
001          LOGICAL  RDDFLG
001          DATA   ASTER/1H*/
004           DATA  ZERO/1H0/, ONE/1H1/, TWO/1H2/, THREE/1H3/
004           DATA  FOUR/1H4/, FIVE/1H5/
001      C
001            MDAY = MCOTAB(1)
001      C
004           MCODE = -1
004           IF  (COMPO.EQ.ONE.OR.COMPO.EQ.TWO.OR.COMPO.EQ.THREE)GOTO 100
004      C       THEN
004      C             WE HAVE A NOTIONAL UNIT SET MCODE TO FIVE
004                    MCODE = FIVE
004                    RETURN
004      C        END-IF
004        100    CONTINUE
004      C
004           IF  (COMPO.NE.TWO .AND. COMPO.NE.THREE)  GOTO 200
004      C       THEN
004      C             WE HAVE EITHER A NATIONAL OR RESERVE UNIT SET MCODE TO FOUR
004                    MCODE = FOUR
004      C        END-IF
004        200    CONTINUE
001      C
001          I = 2
001      C
001      C  DOWHILE   (NOT EOF FOR MCOTAB TABLE)
001      C
001       1000 IF  (I .GT. SIZE0)  GOTO 2000
001      C
001          IF  (ADCO1 .NE. MCOTAB(I))  GOTO 1500
001      C       THEN
001      C             WE HAVE A MATCH FOR MCODE=0 TABLE
001                    MCODE  = ZERO
001                    IF  (.NOT.RDDFLG) ADC03=MDAY
003                     RDDFLG = .TRUE.
001                           RETURN
001      C             END-IF
001      C       ELSE
001       1500             I  =  I + 1
001      C       END-IF
001          GOTO 1000
001      C  END-DO
001      C
001       2000 CONTINUE
004          IF  (MCODE .EQ. FOUR)  RETURN
001          I = 1
001          DECODE (3,1,ROBCO) R1, R2, R3
001        1 FORMAT (3A1)
001      C
001      C  DOWHILE   (NOT EOF FOR MCITAB TABLE)
001      C
001       3000 IF  (I .GT. SIZE1)  GOTO 4000
001          DECODE (3,1,MCITAB(I,1))  T1, T2, T3
001          IF  (T1.NE.ASTER .AND. R1.NE.T1)  GOTO 3100
001      C       THEN
001                    IF  (T2.NE.ASTER .AND. R2.NE.T2)  GOTO 3100
001      C                THEN
```

```
001                  IF   (T3.NE.ASTER .AND. R3.NE.T3)   GOTO 3100
001    C                  THEN
001                          MCODE  =  ONE
001                          IF(.NOT.RDDFLG) ADCO3=MCITAB(I,2+TYPE)
002                          RDDFLG = .TRUE.
001                          RETURN
001    C             END-IF
001    C        END-IF
001    C     ELSE
001     3100       I  =  I + 1
001    C     END-IF
001          GOTO 3000
001    C  END-DO
001    C
001    C
001     4000 CONTINUE
001    C
001          MCODE = THREE
001          RETURN
001          END
```

END ELT.

```
@ELT,L   F5FS.FCKUIC
ELT007 RLIB70 08/16-07:40:21-(2,)
000           FUNCTION CKUIC (FASUIC,TABUIC)
000    C
000          IMPLICIT  INTEGER  (A - Z)
000          DATA  AST/1H*/
000    C
000          DECODE (6,1,FASUIC) F1, F2, F3, F4, F5, F6
000          DECODE (6,1,TABUIC) T1, T2, T3, T4, T5, T6
000        1 FORMAT (6A1)
000    C
000    C     WRITE (6,3)  FASUIC, TABUIC
000     3     FORMAT (1H ,'FUNCTION  FCKUIC  FASUIC=',A6,', TABUIC=',A6)
002          IF(T1.NE.AST.AND.F1.NE.T1)GOTO 2
000    C       THEN
002              IF(T2.NE.AST.AND.F2.NE.T2)GOTO 2
000    C           THEN
002                  IF(T3.NE.AST.AND.F3.NE.T3)GOTO 2
000    C               THEN
002                      IF(T4.NE.AST.AND.F4.NE.T4)GOTO 2
000    C                   THEN
002                          IF(T5.NE.AST.AND.F5.NE.T5)GOTO 2
000    C                       THEN
002                              IF(T6.NE.AST.AND.F6.NE.T6)GOTO 2
000    C                           THEN
000                                    CKUIC = 1
000                                    RETURN
000    C                         END-IF
000    C                     END-IF
000    C                 END-IF
000    C             END-IF
000    C         END-IF
```

```
000  C           ELSE
000      2           CKUIC = 0
000                  RETURN
000  C          END-IF
000          END

END ELT.


@ELT.L   FSFS.MAIN
ELT007 RLIB70 08/16-07:40:22-(29,)
025  C                 F  S  O  R  T   -   R  O  U  T  I  N  E
025  C
025  C                                                    RENE' R. PLOURDE
025  C                                                    USACAA   JUNE 1976
025  C
025  C  THIS ROUTINE PROCESS THE FAS FILE, (FILE 2) A RECORD AT A TIME.  EACH RECORD
025  C  FROM FAS FILE IS WRITTEN TO ONE AND ONLY ONE OF THE OUTPUT FILES (CANDIDATE,
025  C  HAND-PLAYES, LOCKOUT, ALT-THEATER, OR ABOVE-THE-LINE) WITH THE INPUT TABLES,
025  C  (HAND-PLAYES-UNITS (HPUTAB), ALT-THEATER (ALTTAB), LOCKOUT (LOTAB), MCODE=1
025  C  (MC1TAB), MCODE=0 (MC0TAB., AND HAND-PLAYED-COMPO (HPCTAB)) USED TO DETERMINE
025  C  WHICH ONE OF THE OUTPUT FILES TO WRITE THE FAS RECORD TO.
025  C
025        IMPLICIT  INTEGER  (A - Z)
025        LOGICAL  ERCODE, LOFLAG, RDDFLG, ALTFLG
025  C
025  C ***  FILE DEFINITION  ***
025  C
025          DATA  FAS/2/, CFILE/3/, LOFILE/20/, ATLFIL/21/, ALTFIL/22/,
025       *   HPFILE/23/
025          DATA  FILEC/13/, FILELO/31/, FILATL/32/, FILALT/33/, FILEHP/34/
025  C
025  C  DEFINE VARIABL ARRAYS TO INCLUDE THE SORT KEY USED BY THE SYSTEM SORT PROG
025  C
025        INTEGER  RECORD(13), FASREC(52), KEY(241)
026        DATA KEY/1,1,42,0,0,1,2,13,72,0,0,2,99999,228*0/, BLANK/1H /
025        DATA  YES/3HYES/, NO/2HNO/, LEN/100/, N999/3H999/
025  C
025  C  THE ARRAY USES KEY(1) THRU KEY(6) TO DEFINE THE SORT KEY, WHERE
025  C
025  C  KEY(1) = 1  KEY BEGINS IN WORD 1 OF THE RECORD
025  C  KEY(2) = 1  KEY BEGINS IN BIT 1 OF THE WORD SPECIFIED BY KEY(1)
025  C  KEY(3) = 126 TOTAL NUMBER OF BITS USED TO DEFINE THE KEY
025  C  KEY(4) = 0  TYPE OF SORT, ALPHANUMERIC
025  C  KEY(5) = 0  TYPE OF ORDER, ASCENDING
025  C  KEY(6) = 1  OF THE TOTAL NUMBER OF KEYS THIS IS THE FIRST ONE
025  C  KEY(7) = 99999  INDICATES THAT THERE ARE NO MORE KEYS FOR THIS RECORD
025  C
025  C
025        INTEGER        PLIST1(4), PLIST2(19), SIZES(6)
025  C
025  C     DEFINE INPUT TABLES AND THEIR MEANING
025        INTEGER  HPUTAB(100,6), LOTAB(100), ALTTAB(100), MC1TAB(100,3),
025       *         HPCTAB(100), MC0TAB(100)
025  C
025  C     HPUTAB(I,1) =   UIC, HPUTAB(I,2) = ROBCO, HPUTAB(I,3) = ADCO1,
```

```
025   C      HPUTAB(I,4) = ADCO2, HPUTAB(I,5) = ADCO3
025   C      LOTAB(I)    = ADCO1,    ALTTAB(I) = ADCO1,    HPCTAB(I) = COMPO
025   C      MCITAB(I,1) = ROBCO, MCITAB(I,2) = RDD-ATL, MCITAB(I,3) = RDD-BTL
025   C      MCOTAB(1)   = M-DAY, MCOTAB(I) = ADCO1  WHERE I>1
025   C
025   C *** OUTPUT STRUCTURE ***
025   C
025   C      ALL FIVE OUTPUT FILES (CANDIDATE, HAND-PLAYED, LOCKOUT, ALT-THEATER, AND
025   C  ABOVE-THE-LINE) ARE BINARY FILES AND USE THE SAME RECORD STRUCTURE, WHERE
025   C  THE RECORD IS STRUCTURED AS FOLLOWS:   (UICIST  IS THE 1ST CHAR OF THE UIC)
025   C
025   C    SRC15, SRC89, COMPO, MCODE, UICIST, DAMPL, UIC, SEQ, ROBCO, ADCO1, ADCO2,
025   C    ADCO3, STNNM, LOCCO, BRNCH,  UNTDS, TPSN
025   C
025   C    FOR CONVENIENCE WHEN ENCODING THE TWO LIST  PLIST1  AND  PLIST2  ARE USED
025   C
025          EQUIVALENCE (PLIST1(1),SRC15),   (PLIST1(2),SRC89),
025         *            (PLIST1(3),COMPO),   (PLIST1(4),MCODE)
025   C
025          EQUIVALENCE  (PLIST2(1),SRC67),  (PLIST2(2),  SRCA),
025         *             (PLIST2(3),  SRCB),  (PLIST2(4),SRCC),
025         *             (PLIST2(5),BRNCH),  (PLIST2(6),UNTD1),
025         *             (PLIST2(7),UNTD2),  (PLIST2(8),UNTD3),
025         *             (PLIST2(9),UNTD4), (PLIST2(10),TPSN),
025         *            (PLIST2(11),ROBCO), (PLIST2(12),STNM1),
025         *            (PLIST2(13),STNM2), (PLIST2(14),LOCCO),
025         *            (PLIST2(15),SAGR),  (PLIST2(16),ADCO1),
025         *             (PLIST2(17),ADCO2), (PLIST2(18),ADCO3),
025         *             (PLIST2(19),SEQ)
025   C
025           TLKOUT = 0
025           TATL   = 0
025           TALT   = 0
025           THPU   = 0
025           TCAND  = 0
025           SEQ    = 0
025   C
025   C  INITIALIZE  SYSTEM SORT ROUTINE  BY CALLING  SOPEN3
025   C
025          CALL  SOPEN3 ($100,$3000,16,4,KEY)
025   C
025   C  SET-UP-INPUT TABLES BY CALLING SUBROUTINE  SUTAB
025   C
025     100 CALL  SUTAB (HPUTAB,LOTAB,ALTTAB,MCOTAB,MCITAB,HPCTAB,SIZES,
025         *             ERCODE)
025   C
025          IF  (ERCODE)  STOP
025   C
025   C
025   C  DOWHILE  (NOT EOF FOR FAS FILE)
025   C
025    1000 CONTINUE
025   C
025   C    GET NEXT FASREC, DECODE IT, AND INCREMENT SEQ BY ONE
025   C
025          READ (FAS,END=2000)   FASREC
025          DECODE (100,1,FASREC) COMPO, UIC, BRNCH, DAMPL, DSCMP, LOCCO,
```

```
025        .              ADC01, ADC02, ADC03, ROBCO, SRC15, SRC67,
025        . SRC89,SRCA,SRCB,SRCC,STNM1, STNM2, TPSN, TYPCO, UNTD1, UNTD2,
025        .              UNTD3, UNTD4, SAGR
025   C      IF (SEQ .LE. 20)  WRITE(6,7) ADC01,ADC02,ADC03
025   7      FORMAT(' ADC01 = ',A1,'  ADC02 = ',A2,'  ADC03 = ',A3)
025   C
025   C    CHECK FOR HAND-PLAYED-UNITS  THEN CHECK FOR  LOCKOUT-UNITS
025   C
025          CALL CKHPU (UIC,ROBCO,ADC01,ADC02,ADC03,SIZES(1),
025        .             HPUTAB,RDDFLG)
025          CALL CKLO  (ADC01,SRC15,TYPCO,UIC,DSCMP,COMPO,ADC02,ADC03,
025        .             SIZES(2),SIZES(6),LOTAB,HPCTAB,LOFLAG,WHYLO)
025   C
025          IF  (.NOT.LOFLAG)  GOTO 1010
025   C         THEN
025   C               THIS UNIT IS A LOCKOUT UNIT THEREFORE WRITE IT TO THE LOFILE
025                   TLKOUT = TLKOUT + 1
025                   SEQ = TLKOUT
025   C               ENCODE (LEN,2,RECORD) PLIST1, UIC, DAMPL, UIC, PLIST2
025   C               WRITE (LOFILE)  RECORD
025                   MCODE = BLANK
025                   WRITE (FILELO,4) PLIST1, UIC, DAMPL, UIC, PLIST2, WHYLO
025                   GOTO  1990
025   C         ELSE
025   C               CHECK FOR ABOVE-THE-LINE (ATL) UNIT   (TPSN < 20000) = (ATL)
027   1010            CALL  CKALT (ADC01,SIZES(3),ALTTAB,ALTFLG)
027                   IF  (TPSN .GE. 20000)  GOTO 1020
025   C                  THEN
025   C                     THE UNIT IS ABOVE-THE-LINE. THEREFORE CALCULATE THE MCODE
025   C                     FOR THIS UNIT AND THEN WRITE THE RECORD TO ATLFIL.
025                         CALL CMCODE (0,ADC01,ROBCO,ADC03,MCODE,COMPO,
025        .                      SIZES(4),SIZES(5),MC0TAB,MC1TAB,RDDFLG)
025                         TATL = TATL + 1
027                         FLGALT = NO
027                         IF  (ALTFLG)  FLGALT = YES
025                         SEQ = TATL
025                         HPFLAG = NO
025                         IF (RDDFLG) HPFLAG = YES
025                         WRITE (FILATL,4) PLIST1, UIC, DAMPL, UIC, PLIST2,
027        .                                 HPFLAG, FLGALT
025                         GOTO  1980
025   C                  ELSE
025   C                     CHECK FOR ALTERNATE-THEATER  (ALT)
027   1020                  IF  (.NOT.ALTFLG)  GOTO 1030
025   C                        THEN
025   C                           THIS UNIT IS PART OF THE ALT-THEATER THEREFORE
025   C                           WRITE IT TO THE ALTFIL FILE
025                               TALT = TALT + 1
025                               SEQ = TALT
029                               IF (.NOT.RDDFLG) ADC03 = N999
025   C                           ENCODE ( LEN,2,RECORD) PLIST1, UIC, DAMPL,
025   C    .                                                UIC, PLIST2
025   C                           WRITE (ALTFIL)   RECORD
025                               MCODE = BLANK
025                               WRITE (FILALT,4) PLIST1, UIC, DAMPL, UIC,
025        .                                           PLIST2
025                               GOTO 1960
```

```
025   C                        ELSE
025   C                             IS THIS A HAND-PLAYED-UNIT (BELOW-THE-LINE)
025   C                             THAT IS  RDDFLG = .TRUE.
025    1030                                  CALL CMCODE (1,ADC01,ROBCO,
025        *                                               ADC03,MCODE,COMPO,
025        *                                               SIZES(4),SIZES(5),
025        *                                      MCUTAB,MCITAB,RDDFLG)
025                             IF  (.NOT.RDDFLG)  GOTO 1040
025   C                            THEN
025   C                                  THIS UNIT IS A HAND-PLAYED-BELOW-THE-LINE-
025   C                                  UNIT  WRITE IT TO HPFILE
025                                      THPU = THPU + 1
025                                      SEQ = THPU
025   C                                  ENCODE ( LEN,2,RECORD) PLIST1,UIC,
025   C    *                                                DAMPL, UIC, PLIST2
025   C                                  WRITE (HPFILE)   RECORD
025                                      WRITE(FILEHP,4) PLIST1,UIC,DAMPL,
025        *                                            UIC,PLIST2
025                                      GOTO 1960
025   C                            ELSE
025   C                                  PROCESS ONE EACH CANDIDATE RECORD (BTL)
025   C                                  FIRST CHECK SRC POSITIONS 8 AND 9
027    1040                              SAVE89 = SRC89
027                                      CALL CKSRC (SRC15,SRC89)
025                                      SEQ = 0
025                                      DECODE(1,5,UIC) UIC1
025                                      CALL  MODUIC(COMPO,UIC1)
025                                     ENCODE (LEN,2,RECORD) PLIST1, UIC1,
027        *                                    DAMPL, UIC, PLIST2, SAVE89
025                                      CALL   SRREL (RECORD,16)
025                                      TCAND = TCAND + 1
025   C                         END-IF
025    1960                     CONTINUE
025   C                 END-IF
025    1970             CONTINUE
025   C            END-IF
025    1980        CONTINUE
025   C     END-IF
025    1990 CONTINUE
025         GOTO 1000
025   C  END-DO
025   C
025   C  TIME TO SORT THE CANDIDATE RECORDS
025   C
025    2000 CONTINUE
025   C
025         CALL   SSORT
025   C
025   C  NOW GET THE RESULTS OF THE SYSTEM SORT AND WRITE THEM TO THE CANDIDATE FILE
025   C
025   C  DOWHILE  (THERE ARE MORE CANDIDATE RECORDS)
025   C
025    3000 CONTINUE
025           CSEQ = 0
025    3100  CONTINUE
025   C
025         CALL   SRRET (RECORD,DUMMY,$4000)
```

```
025           CSEQ = CSEQ + 1
025     C    WRITE (CFILE)   RECORD
027          DECODE (LEN,2,RECORD) PLIST1, UIC1, DAMPL, UIC, PLIST2, SAVE89
025           SEQ = CSEQ
027          WRITE (FILEC,4) PLIST1, UIC1, DAMPL, UIC, PLIST2, SAVE89
025          GOTO 3100
025     C  END-DO
025     C
025      4000 CONTINUE
025          I  =  1
025     C
025     C  CHECK TO SEE IF ANY HAND-PLAYED-UNITS FROM HPUTAB WERE NOT PLAYED
025     C
025     C  DOWHILE   (NOT EOF FOR HPUTAB)
025     C
025      4100 IF  (I .GT. SIZES(1))  GOTO 5000
025     C
025          IF  (HPUTAB(I,6) .NE. 0)  GOTO 4200
025     C       THEN
025                 WRITE (6,3) I, (HPUTAB(I,J),J=1,6)
025     C    END-IF
025      4200 CONTINUE
025          I  =  I  + 1
025          GOTO 4100
025     C  END-DO
025      5000 CONTINUE
025     C
025     C  PRINT ACCUMULATED TOTAL AND SUB-TOTALS THEN STOP
025     C
025          TOTAL = TLKOUT + TATL + TALT + THPU + TCAND
025          WRITE (6,50)   TLKOUT, TATL, TALT, THPU, TCAND, TOTAL
025       50 FORMAT (1H1,'***  T O T A L S  ***'//
025        * ' LOCKOUT             =',I8/
025        * ' ABOVE-THE-LINE      =',I8/
025        * ' ALT-THEATER         =',I8/
025        * ' HAND-PLAYED-UNITS   =',I8/
025        * ' MATCH-CANDIDATES    =',I8//
025        * ' TOTAL               =',I8)
025     C
025        1 FORMAT (T3,A1,T4,A6,T44,A2,T58,A5,T71,A2,T88,A3/T19,A1,T20,A2,
025        * T22,A3,T30,A3,T34,I5,2A2,3A1,T54,A6,T60,A3,T80,I5,T86,A1/
025        *         T4,A3,T7,A6,T13,A6,T19,A6,T93,A5)
025        2  FORMAT (J5,A2,3A1,A5,A6,A2,3A1,A2,A3,3A6,J5,A3,A6,2A3,A5,
025        *          A1,A2,A3,I6,A6)
025        3 FORMAT (1H0,'HAND-PLAYED-UNIT  ',I4,' WAS NOT USED   UIC = ',
025        * A6,', ROBCO = ',A3,', ADCO1 = ',A1,', ADCO2 = ',A2,', ADCO3 = ',
025        * A3,', FLAG = ',I1)
025        4 FORMAT (1X,J5,A2,3A1,A5,A6,A2,3A1,A2,A3,3A6,J5,A3,A6,2A3,A5,
027        *         A1,A2,A3,I6,A6,A3)
025        5  FORMAT(A1)
025          ENDFILE FILELO
025          ENDFILE FILATL
025          ENDFILE FILALT
025          ENDFILE FILEHP
025          ENDFILE FILEC
025          STOP
025          END
```

```
END ELT.


@ELT,L   FSFS.MODUIC
ELT007 RLIB70 08/16-07:40:24-(0,)
      000   C
      000   C       M  O  D  U  I  C
      000   C
      000   C                                                        RENE' R. PLOURDE
      000   C                                                        USACAA/MRD JUNE 1976
      000   C
      000             SUBROUTINE  MODUIC (COMPO,UICI)
      000   C
      000             IMPLICIT  INTEGER  (A - Z)
      000   C
      000             DATA   K/1HK/,  X/1HX/,  W/1HW/
      000             DATA   ONE/1H1/,  TWO/1H2/,  THREE/1H3/
      000   C
      000             IF  (COMPO .NE. ONE)  GOTO 50
      000   C           THEN
      000                    IF  (UICI .NE. W)  GOTO 10
      000   C                  THEN
      000                           UICI = ONE
      000                           GOTO 40
      000   C                  ELSE
      000        10                 IF  (UICI .NE. X)  GOTO 20
      000   C                         THEN
      000                                  UICI = TWO
      000                                  GOTO 30
      000   C                         ELSE
      000        20                        UICI = THREE
      000   C                       END-IF
      000        30                 CONTINUE
      000   C                END-IF
      000        40          CONTINUE
      000                    GOTO 120
      000   C           ELSE
      000        50          IF  (COMPO.NE.TWO .AND. COMPO.NE.THREE)  GOTO 100
      000   C                  THEN
      000                           IF  (UICI .NE. W)  GOTO 60
      000   C                         THEN
      000                                  UICI = ONE
      000                                  GOTO 90
      000   C                         ELSE
      000        60                        IF  (UICI .NE. K)  GOTO 70
      000   C                                THEN
      000                                         UICI = TWO
      000                                         GOTO 80
      000   C                                ELSE
      000        70                               UICI = THREE
      000   C                              END-IF
      000        80                        CONTINUE
      000   C                       END-IF
      000        90                 CONTINUE
      000                           GOTO 110
      000   C                  ELSE
```

```
000       100          UICI = ONE
000       C                END-IF
000       110          CONTINUE
000       C         END-IF
000       120  CONTINUE
000       C
000            RETURN
000            END

END ELT.


@ELT,L    FSFS.SUTAB
ELT007 RLIB70 08/16-07:40:25-(0,)
000       C                   S U T A B   -   R O U T I N E
000       C
000       C                                                  RENE' R. PLOURDE
000       C                                                  USACAA   JUNE 1976
000       C
000       C     THIS ROUTINE SETS UP THE FOLLOWING SIX TABLES' HAND-PLAYED-UNITS (HPUTAB);
000       C   LOCKOUT-UNITS (LOTAB); ALTERNATE-THEATER (ALTTAB); MCODE=1 (MCITAB);
000       C   MCODE=0 (MCOTAB); AND HAND-PLAYED-COMPOS (HPCTAB).
000       C
000       C
000            SUBROUTINE  SUTAB (HPUTAB,LOTAB,ALTTAB,MCOTAB,MCITAB,HPCTAB,
000          •                   SIZE,ERCODE)
000       C
000            IMPLICIT INTEGER (A-Z)
000            INTEGER  HPUTAB(100,6), LOTAB(100), ALTTAB(100), MCITAB(100,3),
000          •          MCOTAB(100), HPCTAB(100), SIZE(6)
000            LOGICAL   ERCODE
000            DATA   LIMIT1/100/, LIMIT2/100/, LIMIT3/100/,
000          •        LIMIT4/100/, LIMIT5/100/, LIMIT6/100/
000            DATA  HPUNIT/14/, LOUNIT/15/, ALTH/16/,
000          •       MCONE/17/, MCZERO/18/, HPCOMP/19/
000       C
000            ERCODE  =  .FALSE.
000       C
000       C  SET-UP  HAND-PLAYED-UNITS TABLE  (HPUTAB)
000       C
000            N  =  0
000            WRITE (6,101)
000        101 FORMAT (1H1,'•••  HAND-PLAYED-UNITS-TABLE  •••'//1H ,
000          • ' SEQ   UIC      ROBCO   ADC01   ADC02   ADC03'/)
000       C
000       C  DOWHILE   (NOT EOF  AND  NO ERRS)
000       C
000       C
000       C  GET NEXT HAND-PLAYED-UNIT
000       C
000       1000 CONTINUE
000            N = N + 1
000            READ (HPUNIT,1,END=2000)  (HPUTAB(N,J),J=1,5)
000            WRITE (6,11)  N, (HPUTAB(N,J),J=1,5)
000            GOTO  1000
000       C
```

```
000          1 FORMAT (A6,1X,A3,1X,A1,1X,A2,1X,A3)
000         11 FORMAT (1H ,I4,3X,A6,3X,A3,7X,A1,6X,A2,6X,A3)
000      C
000      C  PROCESS ERR FOR TABLE  HPUTAB
000      C
000      C
000         51 FORMAT (1H0,'***  FORMAT ERR  ***')
000      C
000       2000 CONTINUE
000            SIZE(1) = N - 1
000            IF  (SIZE(1) .LE. LIMIT1)  GOTO 2050
000               ERCODE  =  .TRUE.
000               WRITE (6,61)
000         61 FORMAT (1H0,'***  TO MANY INPUT RECORDS  ***')
000       2050 CONTINUE
000            WRITE (6,102)
000        102 FORMAT (1H1,'***  LOCKOUT TABLE  ***'//1H ,' SEQ   ADC01'/)
000            N = 0
000      C  SET-UP LOCKOUT TABLE  (LOTAB)
000      C
000      C
000      C  DOWHILE   (NOT  EOF  AND  NO  ERRS)
000      C
000       2100 CONTINUE
000            N = N + 1
000      C
000            READ (LOUNIT,2,END=3000)   LOTAB(N)
000            WRITE (6,12)  N, LOTAB(N)
000            GOTO 2100
000      C
000          2 FORMAT (A1)
000         12 FORMAT (1H ,I4,5X,A1)
000      C
000      C  PROCESS ERR FOR TABLE  LOTAB
000      C
000      C
000      C
000       3000 CONTINUE
000            SIZE(2) = N - 1
000            IF  (SIZE(2) .LE. LIMIT2)  GOTO 3050
000               ERCODE = .TRUE.
000               WRITE (6,61)
000       3050 CONTINUE
000            WRITE (6,103)
000        103 FORMAT (1H1,'***  ALTERNATE-THEATER-TABLE  ***'//1H ,
000          * ' SEQ   ADC01'/)
000            N = 0
000      C  SET-UP  ALTERNATE-THEATER TABLE  (ALTTAB)
000      C
000      C  DOWHILE   (NOT  EOF  AND  NO  ERRS)
000      C
000       3100 CONTINUE
000            N = N + 1
000      C
000            READ (ALTH,3,END=4000)  ALTTAB(N)
000            WRITE (6,13)  N, ALTTAB(N)
000            GOTO 3100
```

```
      3 FORMAT (A1)
     13 FORMAT (1H ,I4,5X,A1)
C
C
C  PROCESS ERR FOR  ALTERNATE-THEATER INPUT TABLE  (ALTTAB)
C
C
   4000 CONTINUE
        SIZE(3) = N - 1
        IF  (SIZE(3) .LE. LIMIT3)  GOTO 4050
           ERCODE = .TRUE.
           WRITE (6,61)
   4050 CONTINUE
        WRITE (6,104)
    104 FORMAT (1H1,'•••  MCODE = 1 TABLE  •••'//1H ,
      • ' SEQ   ROBCO   RDD/ATL   RDD/BTL'/)
        N = 0
C
C  SET-UP MCODE = 1 TABLE  (MC1TAB)
C
C  DOWHILE   (NOT  EOF  AND  NO ERRS  OCCUR)
C
   4100 CONTINUE
        N = N + 1
C
        READ (MCONE,4,END=5000)  (MC1TAB(N,J),J=1,3)
        WRITE (6,14)  N, (MC1TAB(N,J),J=1,3)
        GOTO 4100
C
      4 FORMAT (A3,2X,A3,2X,A3)
     14 FORMAT (1H ,I4,3X,A3,7X,A3,7X,A3)
C
C  PROCESS ERR FOR  MCODE = 1  TABLE  (MC1TAB)
C
C
   5000 CONTINUE
        SIZE(4) = N - 1
        IF  (SIZE(4) .LE. LIMIT4)  GOTO 5050
           ERCODE = .TRUE.
           WRITE (6,61)
   5050 CONTINUE
        WRITE (6,105)
    105 FORMAT (1H1,'•••  MCODE = 0 TABLE  •••'//1H ,
      • ' SEQ   COMPO'/)
        N = 0
C
C  SET-UP  MCODE = 0 TABLE  (MC0TAB)
C
   5100 CONTINUE
        N = N + 1
C
        READ (MCZERO,5,END=6000)  MC0TAB(N)
        WRITE (6,15)  N, MC0TAB(N)
        GOTO 5100
C
      5 FORMAT (A6)
     15 FORMAT (1H ,I4,5X,A6)
```

```
000   C
000   C  PROCESS  ERR FOR  MCODE = 0 TABLE  (MCOTAB)
000   C
000   C
000    6000 CONTINUE
000         SIZE(5) = N - 1
000         IF  (SIZE(5) .LE. LIMIT5)  GOTO 6050
000            ERCODE = .TRUE.
000            WRITE (6,61)
000    6050 CONTINUE
000         WRITE (6,106)
000      106 FORMAT (1H1,'***  HAND-PLAYED-COMPOS-TABLE  ***'//1H ,
000        * ' SEQ   COMPO'/)
000         N = 0
000   C  SET-UP  HAND-PLAYED-COMPOS TABLE  (HPCTAB)
000   C
000   C  DOWHILE   (NOT  EOF  AND  NO  ERORS  EXIST)
000   C
000    6100 CONTINUE
000         N = N + 1
000   C
000         READ (HPCOMP,6,END=7000)  HPCTAB(N)
000         WRITE (6,16)  N, HPCTAB(N)
000         GOTO 6100
000   C
000        6 FORMAT (A6)
000       16 FORMAT (1H ,I4,3X,A6)
000   C
000   C  PROCESS ERR FOR  HAND-PLAYED-COMPO TABLE  (HPCTAB)
000   C
000   C
000   C
000    7000 CONTINUE
000         SIZE(6) = N - 1
000         IF  (SIZE(6) .LE. LIMIT6)  GOTO 8000
000            ERCODE = .TRUE.
000            WRITE (6,61)
000    8000 CONTINUE
000          CALL CLOSE(HPUNIT,1)
000          CALL CLOSE(ALTH,1)
000          CALL CLOSE(MCONE,1)
000          CALL CLOSE(MCZERO,1)
000          CALL CLOSE(HPCOMP,1)
000   C
000   C  TIME TO RETURN
000   C
000         RETURN
000         END
```

END ELT.

@HDG FMA PROGRAM RSORT **UNCLASSIFIED** .L,0

```
@ELT,L   F5RS.CKSRC
ELT007 RLIB70 08/16-07:40:27-(2,)
     000   C                 C K S R C   -   R O U T I N E
     000   C
     000   C                                                  RENE' R. PLOURDE
     000   C                                                  USACAA/MRD JUNE 1976
     000   C
     000   C     THIS ROUTINE CHECKS SRC POSITIONS 8 AND 9 TO DETERMINE WHETHER OR NOT TO
     000   C   CHANGE THEM TO 99 OR LEAVE UNCHANGED.
     000   C
     000   C
     000           SUBROUTINE  CKSRC (SRC15,SRC89)
     000   C
     000           IMPLICIT  INTEGER  (A - Z)
     000   C
     000           DATA  A/1HA/, Z/1HZ/, N99/2H99/
     000   C
     000           DECODE (2,1,SRC89)  SRC8, SRC9
     000         1 FORMAT(2A1)
     000   C
     001   C       WRITE (6,2) SRC15, SRC89
     000     2     FORMAT(' SUB CKSRC   SRC15=',A5,', SRC89=',A2)
     000   C
     000   C
     002           IF  (SRC15 .EQ. 55018)  GOTO 2000
     002           IF  (SRC15 .EQ. 12067)  GOTO 2000
     000   C
     000           IF  (SRC8.GE.A .AND. SRC8.LE.Z)  GOTO 1000
     000   C           THEN
     000                   SRC89  =  N99
     000                   GOTO 2000
     000   C           ELSE
     000    1000           IF  (SRC9.GE.A .AND. SRC9.LE.Z)  GOTO 2000
     000   C                   THEN
     000                           SRC89  =  N99
     000   C                   ELSE
     000   C                       NO-CHANGE
     000   C               END-IF
     000   C       END-IF
     000    2000 CONTINUE
     000           RETURN
     000           END

END ELT.


@ELT,L   F5RS.RSORT
ELT007 RLIB70 08/16-07:40:27-(2,)
     000   C                 R S O R T   -   R O U T I N E
     000   C
     000   C                                                  RENE' R. PLOURDE
     000   C                                                  USACAA/MRD JUNE 1976
     000   C
     000   C     THIS ROUTINE EXAMINES THE FASTAL REQUIREMENT FILE AND USES THE HAND-PLAYED-
     000   C   UIN FILE, (PROVIDED FOR BY THE USER) TO BUILD A REQUIREMENT FILE.
     000   C
```

```
      IMPLICIT INTEGER  (A-Z)
      INTEGER  RECORD(13), LIST(23), KEY(241), UINTAB(100,3)
      LOGICAL  UINFLG
      EQUIVALENCE  (LIST(1),SRC15),  (LIST(2),SRC89),  (LIST(3),TMPD),
     *             (LIST(4),SRC67),  (LIST(5), SRCA),  (LIST(6),SRCB),
     *             (LIST(7),SRCC),   (LIST(8),UNTD1),  (LIST(9),UNTD2),
     *             (LIST(10),UNTD3), (LIST(11),UNTD4), (LIST(12),RULE),
     *             (LIST(13),LRE),   (LIST(14),COMB),  (LIST(15),ROUND),
     *             (LIST(16),TPSN),  (LIST(17),SOFF),  (LIST(18),SWOF),
     *             (LIST(19),SENL),  (LIST(20),SAGR),  (LIST(21),UIN),
     *             (LIST(22),BRNCH), (LIST(23),STR)
      DATA  LEN/100/,  KEY/1,1,54,0,0,1,99999,234*0/
      DATA  FHPUIN/1/, FILREQ/2/, FILTDA/3/,FASTAL/14/,
     *      FUINHP/11/, REQFIL/12/, TDAFIL/13/, HPUINT/15/
C
C  OPEN SORT ROUTINE
C
      CALL SOPEN3 ($1000,$4000,13,2,KEY)
C
C     READ FIRST TIME PERIOD TO ACCEPT REQUIREMENTS
 1000 READ (HPUINT,4) FIRSTP
      WRITE (6,13)    FIRSTP
C
C     INPUT HAND PLAYED UIN'S
C
      WRITE (6,6)
      SIZE = 1
C
C  DOWHILE   (NOT-EOF-FOR-HPUINT  TABLE)
C
 1050 READ (HPUINT,4,END=1100)  UINTAB(SIZE,1), UINTAB(SIZE,2)
      WRITE (6,5)  SIZE, UINTAB(SIZE,1), UINTAB(SIZE,2)
      SIZE = SIZE + 1
      GOTO 1050
C  END-DO
C
 1100 CONTINUE
      SIZE = SIZE - 1
      SEQ = 1
C
C  DOWHILE   (NOT-EOF-FASTAL-FILE)
C
 2000 READ (FASTAL,1,END=3000) UIN, SRC15, SRC67, SRC89, SRCA, SRCB,
     *  SRCC, BRNCH, UNTD1, UNTD2, UNTD3, UNTD4, ,STR, TMPD, RULE,
     *  LRE, COMB, ROUND, TPSN,  SOFF,  SWOF,  SENL,  SAGR
C     IF(SEQ.LE.100) WRITE(6,3) LIST
      SEQ = SEQ + 1
C
C  CHECK FOR TDA UINTS  (SRC15 IS BLANK)
C
      IF  (SRC15 .EQ. 0)  GOTO 2300
C        THEN
C              CHECK FOR HAND-PLAYED UIN
               UINFLG = .FALSE.
               IF (TMPD .LT. FIRSTP)  UINFLG = .TRUE.
               I = 1
C
```

```
000   C  DOWHILE (NOT END-OF-TABLE  UINTAB)
000   C
000    2100         IF  (UINFLG .OR. I .GT. SIZE)  GOTO 2199
000   C
000                 IF  (UIN .NE. UINTAB(I,1))  GOTO 2150
000                 IF  (TMPD.LT. UINTAB(I,2))  GOTO 2150
000   C                THEN
000                         UINFLG = .TRUE.
000                         UINTAB(I,3) = UINTAB(I,3) + 1
000                         GOTO 2160
000   C                ELSE
000    2150                 I = I + 1
000   C             END-IF
000    2160         CONTINUE
000                 GOTO 2100
000   C  END-DO
000    2199         IF  (.NOT.UINFLG)  GOTO 2200
000   C                THEN
000   C                     THIS RECORD IS A HAND-PLAYED-UIN UNIT
000   C                     IF(SEQ.LE.101) WRITE(6,8)
000   C                     ENCODE (LEN,2,RECORD)  LIST
000   C                     WRITE (FHPUIN)  LIST
000                         WRITE (FUINHP,3)  LIST
000                         THPUIN = THPUIN + 1
000                         GOTO 2250
000   C                ELSE
000   C                     WE HAVE A REQ REOCRD
001    2200                 SAVE89 = SRC89
001                         CALL  CKSRC (SRC15,SRC89)
000   C                     IF(SEQ.LE.101) WRITE(6,9)
001                         ENCODE (LEN,2,RECORD)  LIST, SAVE89
000                         CALL  SRREL (RECORD,13)
000                         TREQ = TREQ + 1
000   C             END-IF
000    2250         CONTINUE
000                 GOTO 2400
000   C          ELSE
000   C             WE HAVE A TDA UNIT
000    2300         CONTINUE
000   C             IF(SEQ.LE.101)  WRITE(6,10)
000   C             ENCODE (32,2,RECORD)  LIST
000   C             WRITE (FILTDA)  LIST
000                 WRITE (TDAFIL,3)  LIST
000                 TTDA = TTDA + 1
000   C     END-IF
000    2400 CONTINUE
000         GOTO 2000
000   C  END-DO
000   C
000    3000 CONTINUE
000   C
000   C  TIME TO SORT REQUIREMENT'S FILE
000   C
000         CALL  SSORT
000   C
000    4000 CONTINUE
000         SEQ = 0
```

```
000    4100  CONTINUE
000            SEQ = SEQ + 1
000          CALL  SRRET (RECORD,MAX,$5000)
001            DECODE (LEN,2,RECORD)  LIST, SAVE89
001            WRITE (REQFIL,3) LIST, SEQ, SAVE89
000          GOTO 4100
000    C
000    C  CHECK UINTAB TO SEE IF ANY HAND-PLAYED UIN'S WERE NOT USED
000    C
000     5000 CONTINUE
000          I = 1
000          WRITE (6,12)
000    C
000    C  DOWHILE    (NOT END OF UNITAB TABLE)
000    C
000     5100 IF  (I .GT. SIZE)  GOTO 6000
000    C
000          WRITE (6,11) I, (UINTAB(I,J),J=1,3)
000     5200 I = I + 1
000          GOTO 5100
000    C  END-DO
000    C
000     6000 CONTINUE
000    C
000    C  COMPUTE TOTAL AND PRINT
000    C
000          TOTAL = THPUIN + TTDA + TREQ
000          WRITE (6,7)  THPUIN, TTDA, TREQ, TOTAL
000        1 FORMAT(I3,2X,I5,2A2,3A1,2X,A2,1X,A3,3A6,12X,I3,1X,I2,2X,A4,
000         * 3(1X,A1),1X,I5,4(1X,A5))
001        2 FORMAT(I5,A2,I2,A2,3A1,A3,3A6,A4,3A1,I5,4A5,I3,A2,I3,A3)
000        3 FORMAT(1X,I5,1X,A2,1X,I2,1X,A2,3(1X,A1),1X,A3,3(1X,A6),1X,A4,
002         * 3(1X,A1),1X,J5,4(1X,A5),1X,I3,1X,A2,1X,I3,I6,A3)
000        4 FORMAT(I3,1X,I2)
000        5 FORMAT(1H ,I4,3X,I3,3X,I2)
000        6 FORMAT(1H ,'***  HAND-PLAYED-UIN-TABLE  ***'//'  SEQ    UIN    TP'/)
000        7 FORMAT(1H1,'***   T O T A L S  ***'//
000         * ' HAND-PLAYED-UIN   =',I8/
000         * ' TDA-UNITS         =',I8/
000         * ' REQUIREMENTS      =',I8//
000         * ' TOTAL             =',I8)
000        8 FORMAT(' ** HAND-PLAYED-UIN **')
000        9 FORMAT(' ** REQ-RECORD **')
000       10 FORMAT(' **  TDA-UNIT  **')
000       11 FORMAT(1H ,I4,3X,I3,3X,I2,3X,I6)
000       12 FORMAT(1H1,'SEQ',3X,'UIC',3X,'TP',3X,'COUNT')
000       13 FORMAT(1H1,'FIRST TIME PERIOD =',I3//)
000          STOP
000          END
```

END ELT.

@HDG   FMA PROGRAM MATCH   **UNCLASSIFIED** .L,0

```
@ELT,L    FSFM.PRINT
ELT007 RLIB70 08/16-07:40:29-(10,)
      006    C
      006    C      P  R  I  N  T    -    R  O  U  T  I  N  E
      006    C
      006             SUBROUTINE  PRT (LINE,TYPE)
      006    C
      006             IMPLICIT INTEGER  (A - Z)
      006             COMMON /FASTAL/ RBRNCH,RSRC15,RSRC67,RSRC89,RSRCA,RSRCB,RSRCC,
      006         *                  RTPSN,RUNTD1,RUNTD2,RUNTD3,RUNTD4,RSOFF,RSWOF,
      006         *                  RSENL,RSAGR,UIN,RULE,TMPD,LRE,UNMBR,OLDSRC,NXUIC,
      009         *          RSTR,REQSEQ,RSTNM1,RSTNM2,RUIC,RLOCCO,RCOMPO,SFR89
      006             COMMON /MCAND/ MSRC15,MSRC67,MSRC89,MSRCA,MSRCB,MSRCC,COMPO,
      006         *                  MC,UIC1ST,DAMPL,UIC,MUNTD1,MUNTD2,MUNTD3,MUNTD4,
      009         *       MCSEQ,MTPSN,MBRNCH,MSAGR,LOCCO,MSTNM1,MSTNM2,ROBCO,SFM89
      006             DATA  RESULT/19/
      006    C
      006    C  ***  CHECK TO SEE IF THE HEADER NEEDS TO BE PRINTED  ***
      006    C
      008             IF  (LINE .LT. 54)  GOTO 500
      006    C          THEN
      006    C               PRINT THE HEADER
      006                    WRITE (RESULT,1)
      006                    LINE = 1
      006                    GOTO 600
      006    C          ELSE
      006    C               INCREMENT LINE COUNTER
      006        500         LINE = LINE + 1
      006    C        END-IF
      006    C
      006        600  CONTINUE
      006    C
      006    C  ***  CHECK FOR SHORT-FALL  (TYPE = 1)  ***
      006    C
      006             IF  (TYPE .NE. 1)  GOTO 1000
      006    C          THEN
      006    C               PRINT ONE EACH SHORT-FALL LINE
      009                    WRITE (RESULT,2) RSRC15,SFR89,RBRNCH,RUNTD1,RUNTD2,
      007         *                RUNTD3,RUNTD4,RCOMPO,RUIC,RTPSN,UIN,TMPD,REQSEQ,
      007         *                          RSTNM1,RSTNM2,RLOCCO,RSTR,RSAGR
      006                    RETURN
      006    C        END-IF
      006    C
      006    C  ***  CHECK FOR A MATCH  (TYPE = 2)  **
      006    C
      006       1000  IF  (TYPE .NE. 2)  GOTO 2000
      006    C          THEN
      006    C               PRINT ONE EACH MATCH LINE
      010                    WRITE (RESULT,3) RSRC15,SFR89,MSRC15,SFM89,MBRNCH,
      006         *                           MUNTD1,MUNTD2,MUNTD3,MUNTD4,COMPO,UIC,
      006         *                           MTPSN,UIN,TMPD,REQSEW,MC,ROBCO,MSTNM1,
      006         *                           MSTNM2,LOCCO,RSTR,MSAGR,MCSEQ
      006                            RETURN
      006    C        END-IF
      006    C
      006    C  ***  CHECK FOR AN OVERAGE  (TYPE - 3)  ***
      006    C
```

```
006   2000 IF  (TYPE .NE. 3)  GOTO 3000
006   C      THEN
006   C           PRINT ONE EACH OVERAGE LINE
009               WRITE (RESULT,4) MSRC15,SFM89,MBRNCH,MUNTD1,MUNTD2,
006        *                         MUNTD3,MUNTD4,COMPO,UIC,MTPSN,MC,
006        *                         ROBCO,MSTNM1,MSTNM2,LOCCO,MSAGR,MCSEQ
006               RETURN
006   C      END-IF
006   C
006   C  ***  ERROR  TYPE .NE. 1, 2, OR 3  ***
006    3000  WRITE (6,99) TYPE
006      99 FORMAT (' *** ERROR TYPE =',I2)
006       1 FORMAT (1H1,'REQ-SRC   FORCE-SRC BRNCH LEVEL UNTDS',14X,
006        * 'COMPO UIC    TPSN  UIN TP RSEQ  MC ROBCO STNNM',5X,
006        * 'LOCCO RSTR  FSTR  FSEQ'//)
006       2 FORMAT (1X,J5,1X,A2,14X,A2,3X,A3,2X,3A6,3X,A1,3X,A6,1X,J5,
006        * 1X,I3,1X,I2,I6,10X,A6,A3,2X,A3,2X,J5,1X,A5)
006       3 FORMAT (1X,J5,1X,A2,2X,J5,1X,A2,4X,A2,3X,A3,2X,3A6,3X,A1,
007        *     3X,A6,1X,J5,1X,I3,1X,I2,1X,I5,1X,A1,3X,A3,2X,A6,A3,
006        *     2X,A3,2X,J5,1X,A5,1X,I5)
006       4 FORMAT (11X,J5,1X,A2,4X,A2,3X,A3,2X,3A6,3X,A1,3X,A6,1X,I5,
006        *  14X,A1,3X,A3,2X,A6,A3,2X,A3,8X,A5,1X,I5)
006         END

END ELT.


@ELT,L   FSFM.SHORT
ELT007 RLIB70 08/16-07:40:30-(23,)
019   C        S H O R T   -   R O U T I N E
019   C
019   C
019         SUBROUTINE  SHORT (NEWSRC)
019   C
019         IMPLICIT  INTEGER  (A - Z)
019         COMMON /TABLES/ NOTAB(20), PTAB(2,23), RODTAB(7,10),
019        *                STP(6,10), KNTROD(10)
019         COMMON /REQREC/ RECORD(25)
019         COMMON /FASTAL/ RBRNCH,RSRC15,RSRC67,RSRC89,RSRCA,RSRCB,RSRCC,
019        *                RTPSN,RUNTD1,RUNTD2,RUNTD3,RUNTD4,RSOFF,RSWOF,
019        *                RSENL,RSAGR,UIN,RULE,TMPD,LRE,UNMBR,OLDSRC,NXUIC,
019        *          RSTR,REQSEQ,RSTNM1,RSTNM2,RUIC,RLOCCO,RCOMPO,SFR89
019         COMMON /MCAND/ MSRC15,MSRC67,MSRC89,MSRCA,MSRCB,MSRCC,COMPO,
019        *               MC,UICIST,DAMPL,UIC,MUNTD1,MUNTD2,MUNTD3,MUNTD4,
019        *     MCSEQ,MTPSN,MBRNCH,MSAGR,LOCCO,MSTNM1,MSTNM2,ROBCO,SFM89
019   C
019         INTEGER  TINDEX(18), RSHORT(47)
019         DATA  TINDEX/1,4,5,6,9,14,15,16,17,18,19,20,21,30,31,32,34,2/
019         DATA ONE/1H1/, FIVE/1H5/, B/1HB/, BLANK/1H /, ASTER/6H******/
021         DATA MCODE9/1H9/,DAMPL9/5H99999/,ZERO/1H0/,RSHORT/47*6H000000/
019   C
019   C  ***  TRANSFER NOTIONAL TABLE INPUT
019   C
019         I = 1
019   C
019   C  DOWHILE  (I .LE. 18)
```

```
019    C
019     1000  IF  (I .GT. 18)  GOTO 1100
019    C
019           K = TINDEX(I)
019           RSHORT(K)  =  NOTAB(I)
019           I = I + 1
019           GOTO 1000
019    C  END-DO
019    C
019    C  ***  TRANSFER FASTAL:S INFO
019    C
019     1100  CONTINUE
019           RSHORT(7)   = UIN
019           RSHORT(8)   = RULE
019           RSHORT(10) = TMPD
019           RSHORT(11) = SFR89
019           RSHORT(12) = LRE
019           RSHORT(13) = RBRNCH
019           RSHORT(22) = BLANK
019           RSHORT(23) = RSRC15
019           RSHORT(24) = RSRC67
019           RSHORT(25) = RSRC89
022           RSHORT(26) = ONE
022           IF  (RSRCA .EQ. FIVE)  RSHORT(26) = B
022           RSHORT(27) = RSRCB
022           RSHORT(28) = RSRCC
019           RSHORT(29) = MCODE9
019           RSHORT(33) = RTPSN
019           RSHORT(37) = RUNTD1
019           RSHORT(38) = RUNTD2
019           RSHORT(39) = RUNTD3
019           RSHORT(40) = RUNTD4
019           RSHORT(41) = RSOFF
019           RSHORT(42) = RSWOF
019           RSHORT(43) = RSENL
019           RSHORT(44) = RSAGR
019           RLOCCO = NOTAB(9)
019           RSTNM1 = NOTAB(15)
019           RSTNM2 = NOTAB(16)
019           RCOMPO = NOTAB(18)
019    C
019    C  ***  COMPUTE UNMBR  ***
019    C
019           IF  (NEWSRC .NE. OLDSRC)  GOTO 1200
019    C        THEN
019                  UNMBR = UNMBR + 1
019                  GOTO 1300
019    C        ELSE
019     1200         UNMBR = 1
019                  OLDSRC = NEWSRC
019    C      END-IF
019     1300  CONTINUE
019           RSHORT(35) = UNMBR
019    C
019    C  ***  LOCATE PECOD FROM PTAB AND TRANSFER IT  ***
019    C
019           I = 1
```

FMA PROGRAM MATCH   **UNCLASSIFIED**   DATE 081676   PAGE 36

```
019            PECOD = ASTER
019            SRC12 = RSRC15 / 1000
019    C
019    C  DOWHILE  (I .LT. 24)
019    C
019     2000  IF  (I .GT. 23)  GOTO 2200
019    C
019            IF  (SRC12 .NE. PTAB(1,I))  GOTO 2100
019    C         THEN
019    C              SRC12 MATCHES PTAB:S SRC12
019                   PECOD = PTAB(2,I)
019                   GOTO 2200
019    C         ELSE
019     2100          I = I  + 1
019    C       END-IF
019            GOTO 2000
019    C  END-DO
019    C
019     2200  CONTINUE
019            RSHORT(36) = PECOD
019    C  ***  SET-UP AUTHORITY FIELDS  ***
019    C
019     3000  CONTINUE
019    C
019    C  ***  COMPUTE UIC36  ***
019    C
019            UIC36  =  NXUIC
019            NXUIC = NXUIC + 1
019             ENCODE(6,2,RSHORT(3)) NOTAB(19), UIC36
019            RUIC = RSHORT(3)
019        2  FORMAT (A2,J4)
019    C
019    C  ***  ENCODE INFO INTO RECORD  ***
019    C
019            ENCODE (150,1,RECORD)  RSHORT
019        1  FORMAT (2A1,2A6,A1,A2,J3,A4,A1,J2,A3,A1,3A2,3A3,A6,A3,A1,A3,
020         *        J5,2A2,4A1,A5,A6,A3,J5,A1,J4,A6,A3,3A6,4A5,A1,A1,A5)
019    C
019    C  ***  INCREMENT TIME-PERIOD COUNTER  ***
019    C
019            KNTRDD(TMPD) = KNTRDD(TMPD) + 1
019    C
019            RETURN
019            END

END ELT.

@ELT,L   FSFM.MAIN
ELT007 RLIB70 08/16-07:40:32-(18,)
014    C
014    C          M  A  I  N   -   R  O  U  T  I  N  E
014    C
014    C                                             RENE: R. PLOURDE
014    C                                             USACAA/MRD  JUNE 1976
014    C
```

```
014          IMPLICIT INTEGER  (A - Z)
014          COMMON /TABLES/ NOTAB(20), PTAB(2,23), RDDTAB(7,10),
014     *                    STP(6,10), KNTRDD(10)
014          COMMON /REQREC/ RECORD(25)
014          COMMON /FASTAL/ RBRNCH,RSRC15,RSRC67,RSRC89,RSRCA,RSRCB,RSRCC,
014     *                    RTPSN,RUNTD1,RUNTD2,RUNTD3,RUNTD4,RSOFF,RSWOF,
014     *                    RSENL,RSAGR,UIN,RULE,TMPD,LRE,UNMBR,OLDSRC,NXUIC,
014     *              RSTR,REQSEQ,RSTNM1,RSTNM2,RUIC,RLOCCO,RCOMPO,SFR89
014          COMMON /MCAND/ MSRC15,MSRC67,MSRC89,MSRCA,MSRCB,MSRCC,COMPO,
014     *                   MC,UICIST,DAMPL,UIC,MUNTD1,MUNTD2,MUNTD3,MUNTD4,
014     *        MCSEQ,MTPSN,MBRNCH,MSAGR,LOCCO,MSTNM1,MSTNM2,ROBCO,SFM89
014          INTEGER  KEY(241), LIST(45), BUFF(24)
014          DATA   LINE/100/
014          DATA  KEY/5,7,12,0,0,1,12,31,6,0,0,2,25,1,36,0,0,3,
014     *                1,13,36,0,0,4,99999,216*0/
014          DATA NXUIC/1/, AUTH13/3H000/, AUTH47/4H0000/, AUTH8/1H0/,
014     *          AUTH9A/2H99/, AUTHBD/3H999/, AUTHE/1H0/, BLANK/1H /
014          DATA  PFILE/11/, RDFILE/12/, NOFILE/13/
014          DATA  FMFILE/1/, FRFILE/2/, OVFILE/3/, MRFILE/8/
014        1 FORMAT (1X,J5,A2,3A1,A5,A6,A2,3A1,A2,A3,3A6,J5,A3,A6,2A3,A5,
014     *        A1,A2,A3,I6,A6)
014        2 FORMAT(1X,I5,1X,A2,1X,I2,1X,A2,3(1X,A1),1X,A3,3(1X,A6),1X,A4,
014     *        3(1X,A1),1X,J5,4(1X,A5),1X,I3,1X,A2,1X,I3,I6,A3)
014        5 FORMAT(I5,A6,A3,A4,A1,A2,A3,A1)
014     C
014          CALL  SOPEN3 ($100,$5000,25,25,KEY)
014     C
014      100 READ (PFILE,11)  PTAB
014       11 FORMAT (I2,2X,A6)
014     C
014          READ (RDFILE,12)  RDDTAB
014       12 FORMAT (7I4)
014     C
014          READ (NOFILE,13)  NOTAB
014       13 FORMAT (A6)
014     C
014     C
014          FLAG = 1
014     C
014     C  DOWHILE  (NOT  EOF)
014     C
014     1000 IF  (FLAG.NE.1 .AND. FLAG.NE.2)  GOTO 1020
014     C       THEN
014     C          GET NEXT MATCH-CANDIDATE-RECORD
014                READ (FMFILE,1,END=2000) MSRC15,MSRC89,COMPO,MC,UICIST,
014     *                                DAMPL,UIC,MSRC67,MSRCA,MSRCB,MSRCC,
014     *                                 MBRNCH,MUNTD1,MUNTD2,MUNTD3,MUNTD4,
014     *                                 MTPSN,ROBCO,MSTNM1,MSTNM2,LOCCO,
014     *                         MSAGR,MADCO1,MADCO2,MADCO3,MCSEQ,SFM89
014               N89 = FLD(0,12,MSRC89)
014             MKEY = MSRC15 * 5000  +  N89
014     C    END-IF
014     C
014     1020 IF  (FLAG.NE.1 .AND. FLAG.NE.3)  GOTO 1040
014     C       THEN
014     C          GET NEXT REQUIREMENT-RECORD
014                READ (FRFILE,2,END=3000) RSRC15,RSRC89,TMPD,RSRC67,RSRCB,
```

```
014        *                     RSRCC,RSRCA,RUNTD1,RUNTD2,RUNTD3,RUNTD4,RULE,
014        *                           LRE,COMB,ROUND,RTPSN,RSOFF,RSWOF,RSENL,
014        *                           RSAGR,UIN,RBRNCH,RSTR,REQSEQ,SFR89
014                  N89 = FLD(0,12,RSRC89)
014                  RKEY = RSRC15 * 5000  +  N89
014    C        END-IF
014    C
014    C   ***  CHECK FOR SHORT-FALL  (RKEY ) MKEY)  ***
014    C
014     1040 IF  (RKEY .GE. MKEY)  GOTO 1060
014    C          THEN
014    C                WE HAVE A SHORT-FALL  CREATE A NOTIONAL UNIT
014                     FLAG = 3
014                     TSHORT = TSHORT + 1
014                     CALL  SHORT (RKEY)
014                    CALL  SRREL (RECORD,25)
014                    CALL  PRT (LINE,1)
014                     GOTO 1200
014    C          ELSE
014    C                ***  CHECK FOR OVERAGE  (RKEY .GT. MKEY)
014     1060            IF  (RKEY .EQ. MKEY)  GOTO 1080
014    C                  THEN
014    C                       WE HAVE AN OVERAGE
014                            FLAG = 2
014                            TOVER = TOVER + 1
014                             CALL  OVER
014                             CALL  SRREL  (RECORD,25)
014                            CALL  PRT (LINE,3)
014                            GOTO 1180
014    C                  ELSE
014    C                       ***  CHECK TIME-PERIOD  ***
014     1080                   IF  (TMPD .EQ. 1)  GOTO 1100
014    C                         THEN
014    C                              WE HAVE A GOOD MATCH
014                                   FLAG = 1
014                                   TMATCH = TMATCH + 1
014                                   CALL  MATCH
014                                   CALL  SRREL (RECORD,25)
014                                   CALL  PRT (LINE,2)
014                                   GOTO 1160
014    C                         ELSE
014    C                              ***  CHECK MCODE FOR 0 OR 1  ***
014     1100                          IF  (MC.NE.0 .AND. MC.NE.1)  GOTO 1120
014    C                                THEN
014    C                                     WE HAVE A GOOD MATCH
014                                          FLAG = 1
014                                          TMATCH = TMATCH + 1
014                                          CALL  MATCH
014                                          CALL  SRREL (RECORD,25)
014                                          CALL  PRT (LINE,2)
014                                          GOTO 1140
014    C                                ELSE
014    C                                     WE HAVE AN IN-COUNTRY SHORT-FALL
014     1120                                 FLAG = 3
014                                          TSHORT = TSHORT + 1
014                                          CALL  SHORT (RKEY)
014                                          CALL  SRREL (RECORD,25)
```

```
014                                 CALL   PRT (LINE,1)
014   C                        END-IF
014    1140                    CONTINUE
014   C                END-IF
014    1160            CONTINUE
014   C        END-IF
014    1180        CONTINUE
014   C     END-IF
014    1200 CONTINUE
014         GOTO 1000
014   C  END-DO
014   C
014   C  •••  PROCESS THE REMAINING REQUIREMENT RECORDS AS SHORT-FALL  •••
014   C
014    2000 IF  (FLAG .EQ. 1)  GOTO 2100
014   C       THEN
014                TSHORT = TSHORT + 1
014                CALL  SHORT (RKEY)
014                CALL  SRREL (RECORD,25)
014                CALL  PRT (LINE,1)
014   C     END-IF
014    2100 CONTINUE
014   C
014   C  DOWHILE   (NOT  EOF  FOR  REQ-FILE)
014   C
014    2200        READ (FRFILE,2,END=4000) RSRC15,RSRC89,TMPD,RSRC67,RSRCB,
014      •                      RSRCC,RSRCA,RUNTD1,RUNTD2,RUNTD3,RUNTD4,RULE,
014      •                            LRE,COMB,ROUND,RTPSN,RSOFF,RSWOF,RSENL,
014      •                      RSAGR,UIN,BRNCH,RSTR,REQSEQ,SFR89
014         RKEY = RSRC15 * 100  +  RSRC89
014         TSHORT = TSHORT + 1
014         CALL  SHORT (RKEY)
014         CALL  SRREL (RECORD,25)
014         CALL  PRT (LINE,1)
014         GOTO 2200
014   C  END-DO
014   C
014   C  •••  PROCESS REMAINING MATCH-CANDIDATE RECORDS AS OVERAGE  •••
014   C
014   C3000  IF  (FLAG .EQ. 1)  GOTO 3100
014   C       THEN
014    3000        TOVER = TOVER + 1
014                 CALL  OVER
014                 CALL  SRREL (RECORD,25)
014                 CALL  PRT (LINE,3)
014   C     END-IF
014    3100 CONTINUE
014   C
014   C  DOWHILE  (NOT  EOF  FOR  MATCH-CAND-FILE)
014   C
014    3200        READ (FMFILE,1,END=4000) MSRC15,MSRC89,COMPO,MC,UIC1ST,
014      •                            DAMPL,UIC,MSRC67,MSRCA,MSRCB,MSRCC,
014      •                             MBRNCH,MUNTD1,MUNTD2,MUNTD3,MUNTD4,
014      •                             MTPSN,ROBCO,MSTNM1,MSTNM2,LOCCO,
014      •                      MSAGR,MADCO1,MADCO2,MADCO3,MCSEQ,SFM89
014         TOVER = TOVER + 1
014                     CALL  OVER
```

```
014                         CALL  SRREL  (RECORD,25)
014                           CALL  PRT (LINE,3)
014            GOTO 3200
014      C  END-DO
014      C
014      C  ***  TIME TO SORT THE MATCH-RESULTS  ***
014      C
014       4000  CALL  SSORT
014      C
014      C  ***  COMPUTE RDD  ***
014      C
014       5000  TP = 1
014              WRITE(6,77) TMATCH,TSHORT,TOVER
014       77     FORMAT(' TMATCH=',I6,', TSHORT=',I6,', TOVER=',I6)
014      C
014      C  DOWHILE  (TP .LT. 10)
014      C
014       5100  IF  (TP .GT. 10)  GOTO 5300
014      C
014             IF  (KNTRDD(TP) .EQ. 0)  GOTO 5200
014      C        THEN
014                 SP = 1
014                  WRITE (6,66)  TP,KNTRDD(TP)
014         66    FORMAT (' TP=',I2,'  KNTRDD(TP)=',I6)
014                 AV = KNTRDD(TP) / 6
014                 R = KNTRDD(TP) - AV * 6
014      C
014      C          DOWHILE  (SP .LT. 7)
014      C
014       5120               IF  (SP .GT. 6)  GOTO 5200
014      C
014                          IF  (SP .GT. R)  GOTO 5140
014      C                     THEN
014                                 L = AV + 1
014                                 GOTO 5160
014      C                     ELSE
014       5140                      L = AV
014      C                   END-IF
014       5160               CONTINUE
014                          M = 1
014      C
014      C                   DOWHILE  (M .LT. L)
014      C
014       5180                          IF  (M .GT. L)  GOTO 5199
014                                     CALL  SRRET (RECORD,DUMMY,$9999)
018                                      DECODE (150,3,RECORD) LIST
014                                      TR = TR + 1
015                                      IF (LIST(45) .NE. '0') GOTO 5182
016                                           TNOT = TNOT + 1
015                                           GOTO 5188
015       5182                           IF (LIST(45) .NE. '1') GOTO 5184
015                                           TMTCH = TMTCH + 1
015                                           GOTO 5188
015       5184                           IF (LIST(45) .NE. '2') GOTO 5186
015                                           TSOV = TSOV + 1
015                                           GOTO 5188
015       5186                            WRITE (6,89) LIST(45)
```

```
015   5188                CONTINUE
017     89                FORMAT (' *** LIST(45) =',A6,' ***')
014                      LIST(25) = LIST(11)
014                        ENCODE (3,88,LIST(11)) RDDTAB(SP,TP)
014     88  FORMAT (J3)
014                      ENCODE (150,3,BUFF) LIST
014                      WRITE (6,1111)  (BUFF(ID),ID=1,12)
014                      WRITE (6,1111)  (BUFF(JD),JD=13,24)
014   1111  FORMAT (20A6)
014                     WRITE(MRFILE)  BUFF
014      3  FORMAT (2A1,2A6,A1,A2,A3,A4,A1,A2,A3,A1,3A2,3A3,A6,A3,A1,A3,
014      *     J5,2A2,4A1,A5,A6,A3,J5,A1,A4,A6,A3,3A6,4A5,A1)
014      4  FORMAT (2A1,2A6,A1,A2,A3,A4,A1,A2,A3,A1,A5,A2,3A3,A6,A3,A1,A5,
014      *   2A2,4A1,2A6,A3,15,A1,14,A6,A3,3A6,4A5,A1)
014                     IF  (LIST(45) .NE. '2')    M = M + 1
014                     GOTO 5180
014   C              END-DO
014   C
014   5199            CONTINUE
014                  WRITE(6,39) TP,SP,M,L,AV,R,TR
014     39      FORMAT(' TP',I3,' SP',I2,' M',I6,' L',I6,' AV',I6,
014      *       ' R',I6,' TR',I6)
014                   SP = SP + 1
014                   GOTO 5120
014   C          END-DO
014   C     END-IF
014   5200  CONTINUE
014     67    FORMAT (' TP=',I2,',  TR=',I6)
014         TP = TP + 1
014         GOTO 5100
014   C  END-DO
014   C
014   9999 CONTINUE
014         IF  (TR .EQ. TMATCH+TSHORT+TOVER)  GOTO 5300
014         WRITE (6,99)
014     99  FORMAT ('1  *** ERROR OCCURRED WHILE COMPUTING RDD ***')
014   5300  CONTINUE
014          WRITE (6,55)  TR, TSOV, TNOT, TMTCH
014     55    FORMAT(' TOTAL NUMBER OF RECORDS RETURNED FROM SORT =',I6,3I6)
014         ENDFILE OVFILE
014         ENDFILE MRFILE
014         STATUS =  CSF(2,'@BRKPT  19  ')
014         STOP
014         END

END ELT.

@ELT,L   FSFM.MATCH
ELT007 RLIB70 08/16-07:40:34-(1,)
000   C
000   C        S  U  B  M  A  T  C  H
000   C
000         SUBROUTINE  MATCH
000   C
000         IMPLICIT  INTEGER  (A - Z)
```

```
000          COMMON /TABLES/ NOTAB(20), PTAB(2,23), RDDTAB(7,10),
000     *                        STP(6,10), KNTRDD(10)
000          COMMON /REQREC/ RECORD(25)
000          COMMON /FASTAL/ RBRNCH,RSRC15,RSRC67,RSRC89,RSRCA,RSRCB,RSRCC,
000     *                        RTPSN,RUNTD1,RUNTD2,RUNTD3,RUNTD4,RSOFF,RSWOF,
000     *                        RSENL,RSAGR,UIN,RULE,TMPD,LRE,UNMBR,OLDSRC,NXUIC,
001     *             RSTR,REQSEQ,RSTNM1,RSTNM2,RUIC,RLOCCO,RCOMPO,SFR89
000          COMMON /MCAND/ MSRC15,MSRC67,MSRC89,MSRCA,MSRCB,MSRCC,COMPO,
000     *                       MC,UICIST,DAMPL,UIC,MUNTD1,MUNTD2,MUNTD3,MUNTD4,
001     *         MCSEQ,MTPSN,MBRNCH,MSAGR,LOCCO,MSTNM1,STNM2,ROBCO,SFM89
000          INTEGER  RMATCH(47)
000          DATA  RMATCH/44*6H        ,1H1,2*1H /
000          DATA  ONE/1H1/
000     C
000     C  ***  SET-UP THE AUTHORITY FIELDS IN RMATCH  ***
000     C
000          RMATCH(2) = COMPO
000          RMATCH(3) = UIC
000          RMATCH(7) = UIN
000          RMATCH(8) = RULE
000          RMATCH(9) = ONE
000          RMATCH(10) = TMPD
001          RMATCH(11) = SFM89
000          RMATCH(12) = LRE
000          RMATCH(22) = ROBCO
000          RMATCH(29) = MC
000          RMATCH(46) = UICIST
000          RMATCH(47) = DAMPL
000     C
000     C  ***  ENCODE INFO INTO RECORD  ***
000     C
000          ENCODE (150,1,RECORD) RMATCH
000     C
000     C  ***  INCREMENT TIME-PERIOD COUNTER  ***
000     C
000          KNTRDD(TMPD) = KNTRDD(TMPD) + 1
000     C
000          RETURN
000        1 FORMAT (2A1,2A6,A1,A2,J3,A4,A1,J2,A3,A1,3A2,3A3,A6,A3,A1,A3,
000     *          A5,2A2,4A1,A5,A6,A3,A5,A1,A4,A6,A3,3A6,4A5,A1,A1,A5)
000          END

END ELT.

@ELT,L   FSFM.OVER
ELT007 RLIB70 08/16-07:40:35-(2,)
000     C
000     C      O  V  E  R  A  G  E
000     C
000          SUBROUTINE  OVER
000     C
000          IMPLICIT  INTEGER (A-Z)
000     C
000          COMMON /TABLES/ NOTAB(20), PTAB(2,23), RDDTAB(7,10),
000     *                        STP(6,10), KNTRDD(10)
```

```
000          COMMON /REQREC/ RECORD(25)
000          COMMON /FASTAL/ RBRNCH,RSRC15,RSRC67,RSRC89,RSRCA,RSRCB,RSRCC,
000      *                   RTPSN,RUNTD1,RUNTD2,RUNTD3,RUNTD4,RSOFF,RSWOF,
000      *                   RSENL,RSAGR,UIN,RULE,TMPD,LRE,UNMBR,OLDSRC,NXUIC,
001      *           RSTR,REQSEQ,RSTNM1,RSTNM2,RUIC,RLOCCO,RCOMPO,SFR89
000          COMMON /MCAND/ MSRC15,MSRC67,MSRC89,MSRCA,MSRCB,MSRCC,COMPO,
000      *                  MC,UICIST,DAMPL,UIC,MUNTD1,MUNTD2,MUNTD3,MUNTD4,
001      *       MCSEQ,MTPSN,MBRNCH,MSAGR,LOCCO,MSTNM1,MSTNM2,ROBCO,SFM89
000    C
000          INTEGER  OVLIST(47)
000          DATA  MCODE8/1H8/
000    C
000          DATA  OVLIST/9*6H000000,2*6H999999,33*6H000000,1H2,2*1H /
000          OVLIST(22) =  ROBCO
000          OVLIST(10) =  9
000          OVLIST(29) =  MCODE8
000          OVLIST(46) =  UICIST
000          OVLIST(47) =  DAMPL
000          OVLIST(2) =  COMPO
000          OVLIST(3) =  UIC
000          KNTRDD(9) = KNTRDD(9) + 1
000    C
000          ENCODE (150,1,RECORD)  OVLIST
000    C
000    C   D I T T O       FORMAT 1 FROM SUBMATCH ALIAS MATCH
002        1 FORMAT (2A1,2A6,A1,A2,A3,A4,A1,J2,A3,A1,3A2,3A3,A6,A3,A1,A3,
000      *           A5,2A2,4A1,A5,A6,A3,A5,A1,A4,A6,A3,3A6,4A5,A1,A1,A5)
000    C      RETURN
000           RETURN
000           END
```

END ELT.

@HDG FMA PROGRAM ATL **UNCLASSIFIED** .L.0

```
@ELT,L    FSATL.FIND
ELT007 RLIB70 08/16-07:40:37-(0,)
000   C
000         SUBROUTINE  FIND (KEY,I,SIZE)
000         IMPLICIT  INTEGER  (A - Z)
000         COMMON /TTAB/  TABLE(5,100)
000   C
000         MAX = SIZE
000         MIN  = 1
000         I = 0
000   C
000   C  DOWHILE    (I .EQ. 0  .AND.  MIN .LE. MAX)
000   C
000    1000  IF  (I.NE.0 .OR. MIN.GT.MAX)  RETURN
000   C
000         MID = (MAX + MIN)/2
000         IF  (KEY .NE. TABLE(1,MID))  GOTO  1100
000   C              WE HAVE FOUND THE KEY
000                  I  =  MID
000                  TABLE(5,I)  =  1
000                  GOTO 1200
000   C          ELSE
000    1100          IF  (KEY .LT. TABLE(1,MID))  GOTO 1120
000   C                THEN
000   C                      POSSIBLE MATCH IS IN UPPER HALF OF TABLE
000                          MIN = MID + 1
000                          GOTO 1140
000   C                ELSE
000   C                      POSSIBLE MATCH IS IN LOWER HALF OF TABLE
000    1120                  MAX = MID - 1
000   C              END-IF
000    1140          CONTINUE
000   C      END-IF
000    1200  CONTINUE
000         GOTO 1000
000   C  END-DO
000   C
000         END

END ELT.
```

```
@ELT,L    FSATL.GETTP
ELT007 RLIB70 08/16-07:40:37-(0,)
000   C
000         SUBROUTINE GETTP (RDD,TP)
000   C
000         IMPLICIT  INTEGER (A - Z)
000         COMMON /RTAB/  RDDTAB(7,10)
000   C
000         I = 1
000   C
000   C  DOWHILE    (I .LE. 10)
000   C
000    1000  IF  (I .GT. 10)  GOTO 2000
000   C
```

```
000          IF  (RDD .GT. RDDTAB(7,I))  GOTO 1100
000    C       THEN
000    C            WE HAVE FOUND THE CORRECT TIME-PERIOD (TP)
000                 TP = I
000                 RETURN
000    C       ELSE
000     1100        I = I + 1
000    C     END-IF
000          GOTO 1000
000    C  END-DO
000    C
000     2000 CONTINUE
000          WRITE (6,1)  RDD
000        1 FORMAT ('0  ••• RDD = ',I3,', WAS NOT FOUN IN RDDTAB  •••'//)
000          TP = 0
000          RETURN
000          END
```

END ELT.

```
@ELT,L   FSATL.MAIN
ELT007 RLIB70 08/16-07:40:39-(3,)
000    C
000    C      A  B  O  V  E  -  T  H  E  -  L  I  N  E
000    C
000    C                                            RENE' R. PLOURDE
000    C                                            USACCA/MRD  JUNE 1976
000    C
000          IMPLICIT  INTEGER  (A - Z)
000          COMMON /TTAB/  TABT(5,100)
000          COMMON /RTAB/  RDDTAB(7,10)
000          INTEGER  LIST(27)
000          EQUIVALENCE (LIST(1),SRC15), (LIST(2),SRC89), (LIST(3),COMPO),
000       •            (LIST(4),   MC), (LIST(5),UICIST), (LIST(6),DAMPL),
000       •            (LIST(7),  UIC), (LIST(8),SRC67), (LIST(9), SRCA),
000       •           (LIST(10), SRCB),(LIST(11), SRCC),(LIST(12),BRNCH),
000       •           (LIST(13),UNTD1),(LIST(14),UNTD2),(LIST(15),UNTD3),
000       •           (LIST(16),UNTD4),(LIST(17),TPSN ),(LIST(18),ROBCO),
000       •           (LIST(19),STNM1),(LIST(20),STNM2),(LIST(21),LOCCO),
000       •           (LIST(22), SAGR),(LIST(23),ADCO1),(LIST(24),ADCO2),
000       •           (LIST(25),ADCO3),(LIST(26),  SEQ),(LIST(27),HPFLAG)
000          DATA RDDFIL/11/, TPSNF/12/, ATLFIL/13/, FILATL/14/
000          DATA  YES/3HYES/, NO/2HNO/
000          DATA  ZERO/1H0/, ONE/1H1/, BLANK/6H      /, ASTER/1H•/
000    C
000    C
000    C  •••  GET RDD TABLE FROM RDDFIL FILE  •••
000    C
000          I = 1
000    C
000    C  DOWHILE    (I .LE. 10)
000    C
000     1000 IF  (I .GT. 10)  GOTO  1999
000    C
000          READ (RDDFIL,1)  (RDDTAB(J,I),J=1,7)
```

```
000            WRITE (6,2)  I, (RDDTAB(J,I),J=1,7)
000            I = I + 1
000            GOTO 1000
000    C  END-DO
000    C
000     1999   CONTINUE
000            I = 1
000    C
000    C  ***  LOAD TPSN TABLE FROM TPSNF FILE TO TABT TABLE  ***
000    C
000    C  DOWHILE    (NOT EOF  FOR TPSNF)
000    C
000     2000   READ (TPSNF,3,END=2999)  (TABT(J,I),J=1,3)
000            CALL  GETTP (TABT(3,I),TABT(4,I))
000            WRITE (6,4)  I, (TABT(J,I),J=1,4)
000            I = I + 1
000            GOTO 2000
000    C  END-DO
000    C
000     2999   CONTINUE
000            TSIZE = I - 1
000    C
000    C  ***  PROCESS ATL RECORDS  ***
000    C
000    C  DOWHILE    (NOT EOF FOR ATLFIL)
000    C
001     3000   READ (ATLFIL,5,END=3999)  LIST, FLGATL
000            CALL  FIND (TPSN,INDEX,TSIZE)
000            IF  (INDEX .EQ. 0)  GOTO 3300
000    C         THEN
000    C                 ***  ADCO1 = 1  ***
002                      ADCO1 = TABT(2,INDEX)
000                      LOFLAG = BLANK
000                      IF  (TABT(2,INDEX) .NE. ONE)  GOTO3100
000    C                   THEN
000    C                         ***  HPFLAG  ***
003                            IF  (HPFLAG.EQ.YES .AND. ADCO3.NE.BLANK)  GOTO 3020
000    C                         THEN
000                                    ADCO2 = TABT (4,INDEX)
000                                    ADCO3 = TABT(3,INDEX)
000                                    GOTO 3040
000    C                         ELSE
000    C                               COMPUTE NEW TP BASED ON INPUT ADCO3
003     3020                       DECODE (3,7,ADCO3) ADCO3
003                                CALL  GETTP (ADCO3,ADCO2)
000    C                       END-IF
000     3040                   CONTINUE
000                            GOTO 3200
000    C                 ELSE
000     3100                   ADCO2 = 0
003                            IF  (HPFLAG.EQ.YES .AND. ADCO3.NE.BLANK)  GOTO 3120
000    C                         THEN
000                                    ADCO3 = TABT(3,INDEX)
003                                    GO TO 3200
000    C                       END-IF
003     3120                   DECODE (3,7,ADCO3) ADCO3
000    C                 END-IF
```

```
000    3200          CONTINUE
000                  GOTO 3400
000    C          ELSE
000    3300          LOFLAG = ASTER
000    C       END-IF
000     3400   CONTINUE
000                 LIST(27) = LOFLAG
003            WRITE (FILATL,8)  LIST, FLGATL
000            GOTO 3000
000    C   END-DO
000    C
000     3999   CONTINUE
000            I = 1
000    C
000    C   DOWHILE     ( I .LE. TSIZE)
000    C
000     4000   IF   (I .GT. TSIZE)   GOTO 4999
000    C
000            IF   (TABT(5,1) .NE. 0)   GOTO 4100
000    C          THEN
000    C             PRINT THAT THIS TPSN WAS NOT USED
000                  WRITE (6,6)   I, (TABT(J,I),J=1,5)
000    C       END-IF
000     4100   CONTINUE
000            I = I + 1
000            GOTO 4000
000    C   END-DO
000    C
000     4999   CONTINUE
000    C
000            WRITE (6,101)
000      101   FORMAT(' *** ABOVE-THE-LINE  FINI ***')
000            STATUS = CSF (2,'@BRKPT  11  ')
000            STATUS = CSF (2,'@BRKPT  12  ')
000            STATUS = CSF (2,'@BRKPT  13  ')
000            STATUS = CSF (2,'@BRKPT  14 ')
000            STOP
000        1   FORMAT (7I4)
000        2   FORMAT (1X,I3,7I6)
000        4   FORMAT (1X,I3,2X,J5,3X,A1,3X,J3,2X,J2)
000        3   FORMAT (J5,2X,A1,2X,J3)
000        5   FORMAT (1X,J5,A2,3A1,A5,A6,A2,3A1,A2,A3,3A6,J5,A3,A6,2A3,A5,
003       *             A1,J2,A3,I6,A6,A3)
000        6   FORMAT (1H0,I3,1X,'*** TPSN=',J5,' WAS NOT USED ***',2X,A1,
000       *           J3,2X,J2,2X,I2)
003        7   FORMAT (J3)
003        8   FORMAT (1X,J5,A2,3A1,A5,A6,A2,3A1,A2,A3,3A6,J5,A3,A6,2A3,A5,
003       *             A1,J2,J3,I6,A6,A3)
000            END
```

END ELT.

@HDG   FMA PROGRAM ALT   ••UNCLASSIFIED•• .L.U

```
@ELT,L   FSALT.MAIN
ELT007 RLIB70 08/16-07:40:40-(1,)
        000    C
        000    C        A  L  T  E  R  N  A  T  E  -  T  H  E  A  T  E  R
        000    C
        000    C                                                        RENEQ R. PLOURDE
        000    C                                                        USACCA/MRD  JUNE 1976
        000    C
        000              IMPLICIT  INTEGER  (A - Z)
        000              INTEGER  ALTAB(6,100),  RDD(100), LIST(26)
        000              DATA  ALTTF/11/, ALTFIL/12/, FILALT/13/
        000    C
        000              EQUIVALENCE (LIST(1),SRC15), (LIST(2),SRC89), (LIST(3),COMPO),
        000         •                (LIST(4),    MC), (LIST(5),UICIST), (LIST(6),DAMPL),
        000         •                (LIST(7),   UIC), (LIST(8),SRC67), (LIST(9), SRCA),
        000         •               (LIST(10), SRCB),(LIST(11), SRCC),(LIST(12),BRNCH),
        000         •               (LIST(13),UNTD1),(LIST(14),UNTD2),(LIST(15),UNTD3),
        000         •               (LIST(16),UNTD4),(LIST(17),TPSN ),(LIST(18),ROBCO),
        000         •               (LIST(19),STNM1),(LIST(20),STNM2),(LIST(21),LOCCO),
        000         •               (LIST(22), SAGR),(LIST(23),ADCO1),(LIST(24),ADCO2),
        000         •               (LIST(25),ADCO3),(LIST(26),  SEQ)
        000    C
        000              WRITE (6,21)
        000          21  FORMAT ('1ADCO1  LOWER-LIMIT  UPPER-LIMIT  INCRE  NEW-ADCO1'//)
        000          11  FORMAT (3X,A1,8X,J3,10X,J3,7X,J3,7X,A1)
        000              I = 1
        000    C
        000    C  DOWHILE    (NOT EOF FOR ALTTF)
        000    C
        000     1000  READ (ALTTF,1,END=1999)  (ALTAB(J,I),J=1,5)
        000        1  FORMAT (A1,2X,J3,2X,J3,2X,J3,2X,A1)
        000              WRITE (6,11)  (ALTAB(J,I),J=1,5)
        000              RDD(I) = ALTAB(2,I) - ALTAB(4,I)
        000              I = I + 1
        000              GOTO  1000
        000    C  END-DO
        000    C
        000     1999  CONTINUE
        000              TSIZE = I - 1
        000    C
        000    C  DOWHILE    (NOT EOF FOR ALTFIL)
        000    C
        000     2000  READ (ALTFIL,8,END=2999)  LIST
        000        8  FORMAT (1X,J5,A2,3A1,A5,A6,A2,3A1,A2,A3,3A6,J5,A3,A6,2A3,A5,
        001         •           A1,J2,J3,I6,A6)
        000              I = 1
        000              J = 0
        000    C
        000    C         DOWHILE    (I .LE. TSIZE  .AND.  J .EQ. 0)
        000    C
        000     2100           IF  (I.GT.TSIZE .OR. J.NE.0)  GOTO  2199
        000    C
        000                    IF  (ADCO1 .NE. ALTAB(1,I))  GOTO 2120
        000    C                 THEN
        000    C                     ADCO1'S MATCH
        000                          J = I
        000                          RDD(J) = RDD(J) + ALTAB(4,J)
```

```
000      IF (RDD(J) .GT. ALTAB(2,J))  RDD(J) = ALTAB(2,J)
000      ADC01  =  ALTAB(5,J)
000      IF  (ADC03 .EQ. 999)  ADC03 = RDD(J)
000        ADC02 = 0
000      ALTAB(6,J) = 1
000      GOTO 2140
000  C            ELSE
000   2120           I = I + 1
000  C          END-IF
000   2140          CONTINUE
000                 GOTO 2100
000  C      END-DO
000  C
000   2199  CONTINUE
000  C
000         IF  (J .NE. 0)  GOTO 2200
000  C        THEN
000  C             ADC01 WAS NOT FOUND
000                WRITE (6,2)  ADC01
000       2 FORMAT (1H0,'***  ADC01 = ',A1,' WAS NOT FOUND IN THE TABLE ***')
000                LOFLAG = 1
000                GOTO 2220
000  C        ELSE
000   2200         WRITE (FILALT,8)  LIST
000  C      ENDIF
000   2220  CONTINUE
000         GOTO 2000
000  C  END-DO
000  C
000   2999  CONTINUE
000         WRITE (6,101)
000    101  FORMAT (' *** ALTERNATE-THEATER FINI ***')
000  C
000         STATUS  =  CSF (2,'@BRKPT  11  ')
000         STATUS  =  CSF (2,'@BRKPT  12  ')
000         STATUS  =  CSF (2,'@BRKPT  13  ')
000         STOP
000         END
```

END ELT.

@HDG   FMA PROGRAM LAYIN   ••UNCLASSIFIED•• .L,U

```
@ELT,L   FSLAY.MAIN
ELT007 RLIB70 08/16-07:40:42-(0,)
   000   C
   000   C        L   A   Y   I   N
   000   C
   000   C                                                      RENEW R. PLOURDE
   000   C                                                      USACAA/MRD  JUNE 1976
   000   C
   000            IMPLICIT  INTEGER  (A - Z)
   000            COMMON  /RTAB/  RDDTAB(7,10)
   000            INTEGER  KEY(241), NOTREC(52), FASREC(52)
   000            INTEGER  ALTTAB(5), ATLTAB(8), LOTAB(6), HPTAB(6), MRTAB(45)
   000            INTEGER  NINDEX(44), MRREC(24)
   000   C
   000            INTEGER  FASTAB(99), NOTTAB(99)
   000            DATA  KEY/1,7,42,0,0,1,99999,234*0/
   000            DATA  FAS/3HFAS/, NOT/3HNOT/, ZERO/6H000000/, ONE/6H111111/
   000            DATA  COMPO/5HCOMPO/, X/1HX/, NINES/6H999999/
   000            DATA  ALTFIL/11/, ATLFIL/12/, LOFILE/13/, HPFILE/14/
   000            DATA  MRFILE/15/, FASFIL/16/, FILEF/17/, RDDFIL/10/
   000            DATA  (NOTTAB(J),J=1,75)/75*6H      /
   000            DATA  NOTTAB(98)/6H      /, NOTTAB(99)/6H      /
   000            DATA  (NOTTAB(K),K=83,97)/15*6H000000/
   000            DATA  NEXT/3HFAS/, ALTUIC/-1/, ATLUIC/-1/, LOUIC/-1/, HPUIC/-1/
   000            DATA  MUIC/-1/, BLANK/1H /
   000            DATA  NINDEX/1,2,3,4,5,8,9,10,11,12,13,14,15,16,24,31,
   000       $                33,36,38,39,50,53,55,56,57,58,59,60,61,62,64,65,
   000       $                70,72,76,77,78,79,80,81,93,94,95,96/
   000            EQUIVALENCE  (ALTTAB(1),ALTUIC),  (ATLTAB(2),ATLUIC)
   000            EQUIVALENCE  (LOTAB(1),LOUIC),  (HPTAB(2),HPUIC)
   000            EQUIVALENCE  (MRTAB(3),MUIC),  (FASTAB(3),FASUIC)
   000            EQUIVALENCE  (ATLTAB(7),ATLFLG),  (LOTAB(6),LOTYPE)
   000            EQUIVALENCE  (MRTAB(45),MRFLAG),  (ATLTAB(8),FLGALT)
   000            EQUIVALENCE  (FASTAB(9),AUTH13), (FASTAB(10),AUTH47)
   000            EQUIVALENCE  (FASTAB(11),AUTH8), (FASTAB(12),AUTH9A)
   000            EQUIVALENCE  (FASTAB(13),AUTHBD), (FASTAB(14),AUTHE)
   000            EQUIVALENCE  (FASTAB(53),ROBCO)
   000   C
   000        1   FORMAT (1X,2A1,2A6,A1,A6,A4,A2,A3,A4,A1,A2,A3,3A2,A6,2A2,
   000       *           A5,A3,A2,A3,2A2,5A1,A3,A5,2A3,A2,A3,A2,A6,A3,A2,
   000       *           A1,A2,A1,A3,A1,A2,A3,5A1,A3,A1,A5,2A2,4A1,A5,A2,
   000       *           A6,A3,A5,A6,A5,A1,A5,4A1,A5,A4,A6,A3,3A6,A3,15A5,
   000       *           A4,A1)
   000        3   FORMAT (2A1,2A6,A1,A2,A3,A4,A1,A2,A3,A1,3A2,3A3,A6,A3,
   000       *           A1,A3,A5,2A2,4A1,A5,A6,A3,A5,A1,A4,A6,A3,3A6,
   000       *           4A5,A1)
   000   C
   000        2   FORMAT (16X,A6,33X,A3,17X,A1,A2,A3,6X,A6)
   000   C
   000   C
   000            CALL  SOPEN3 ($100,$2000,24,2,KEY)
   000   C
   000      100   CONTINUE
   000   C
   000   C
   000   C  •••  GET RDD TABLE FROM RDDFIL FILE  •••
   000   C
```

```
000         I = 1
000   C
000   C  DOWHILE    (I .LE. 10)
000   C
000    150  IF  (I .GT. 10)  GOTO  199
000   C
000         READ (RDDFIL,6)  (RDDTAB(J,I),J=1,7)
000         WRITE (6,7)  I, (RDDTAB(J,I),J=1,7)
000         I = I + 1
000         GOTO 150
000   C  END-DO
000   C
000    199   CONTINUE
000      6  FORMAT (7I4)
000      7  FORMAT (1X,I3,7I6)
000   C
000   C  DOWHILE    (NOT  EOF  FOR  MRFILE)
000   C
000   1000  READ (MRFILE,END=1999)  MRREC
000         CALL  SRREL (MRREC,24)
000         GOTO 1000
000   C  END-DO
000   C
000   1999  CONTINUE
000   C
000         CALL  SSORT
000   C
000   C
000   2000  CONTINUE
000   C
000         READ (ALTFIL,2,END=2100)  ALTTAB
000   C
000   2100  READ (ATLFIL,8,END=2200)  ATLTAB
000   C
000   2200  READ (LOFILE,2,END=2300)  LOTAB
000   C
000   2300  READ (HPFILE,8,END=2400)  HPTAB
000   C
000   2400  CALL  SRRET (MRREC,DUMMY,$2500)
000         DECODE (150,3,MRREC)  MRTAB
000   C
000   2500  CONTINUE
000   C
000   C  ***  MERGE THE SIX INPUT FILES TO PRODUCE THE OUTPUT FILE  ***
000   C
000   C
000   C  DOWHILE    (NOT  EOF  FOR  FAS-FILE)
000   C
000   3000  IF  (NEXT .NE. FAS)  GOTO 3100
000   C        THEN
000   C              GET NEXT FAS RECORD
000               LUIC = FASUIC
000               READ (FASFIL,END=3600)  FASREC
000               DECODE (312,1,FASREC)  FASTAB
000   C        END-IF
000   3100  CONTINUE
000         IF  (NEXT.EQ.NOT .AND. MUIC.EQ.-1)  GOTO 3999
```

```
000            COUNT = COUNT + 1
000    C
000    C  ***  COMPARE  ALT-TH (UIC)  WITH  FAS (UIC)  ***
000    C
000            IF  (ALTUIC .NE. FASUIC)  GOTO 3200
000    C         THEN
000    C               PROCESS ONE EACH  ALT-TH  WITH THE CURRENT  FAS-REC
000                    TALT = TALT + 1
000                    ROBCO = ALTTAB(2)
000                    AUTH8 = FASTAB(45)
000                    AUTH9A = BLANK
000                    AUTHBD = ALTTAB(5)
000                    ENCODE (2,102,AUTHE) BLANK, ALTTAB(3)
000                    AUTH13 = BLANK
000                    AUTH47 = BLANK
000                    NEXT = FAS
000                   IF  (MOD(COUNT,50) .EQ. 0) CALL  PRTFAS (COUNT,1,FASTAB)
000                    ENCODE (312,1,FASREC)  FASTAB
000                    WRITE (FILEF)  FASREC
000                    READ (ALTFIL,2,END=3199)  ALTTAB
000    C       END-IF
000            GOTO 3990
000     3199   ALTUIC = -1
000            GOTO 3990
000    C
000    C  ***  COMPARE  ABOVE-THE-LINE (UIC)  WITH  FAS (UIC)  ***
000    C
000     3200   IF  (ATLUIC .NE. FASUIC)  GOTO 3400
000    C         THEN
000    C               CHECK FOR ATL-OVERAGE
000                    IF  (ATLFLG .NE. '1')  GOTO 3300
000    C                 THEN
000    C                      TREAT THIS RECORD AS AN OVERAGE
000                           TATLOV = TATLOV + 1
000                           AUTH13 = ZERO
000                           AUTH47 = ZERO
000                           AUTH8  = ZERO
000                           AUTH9A = NINES
000                           AUTHBD = NINES
000                           AUTHE  = ZERO
000                   IF  (MOD(COUNT,50) .EQ. 0) CALL  PRTFAS (COUNT,2,FASTAB)
000                           GOTO 3350
000    C                 ELSE
000    C                      PROCESS ONE EACH ATL WITH CURRENT FAS REC
000     3300                  TATL = TATL + 1
000                           ROBCO = ATLTAB(3)
000                           AUTH13 = BLANK
000                           AUTH47 = BLANK
000                           AUTH8  = ONE
000                           AUTH9A = ATLTAB(5)
000                           AUTHBD = ATLTAB(6)
000                           LRE  = ONE
000                           IF  (FLGALT .NE. YES)  GOTO 3325
000                             AUTH8 = FASTAB(45)
000                             AUTH9A = BLANK
000                             LRE = BLANK
000     3325                  CONTINUE
```

```
000                    ENCODE (2,102,AUTHE) LRE, ATLTAB(4)
000               IF  (MOD(COUNT,50) .EQ. 0) CALL  PRTFAS (COUNT,3,FASTAB)
000   C            END-IF
000    3350        NEXT = FAS
000                ENCODE (312,1,FASREC)  FASTAB
000                WRITE (FILEF)  FASREC
000                READ (ATLFIL,8,END=3399)  ATLTAB
000   C       END-IF
000           GOTO 3980
000    3399   ATLUIC = -1
000           GOTO 3980
000   C
000   C  •••  COMPARE  LOCKOUT (UIC)  WITH  FAS (UIC  •••
000   C
000    3400   IF  (LOUIC .NE. FASUIC)  GOTO  3500
000   C         THEN
000   C            CHECK LOCKOUT TYPE
000                IF  (LOTYPE .NE. COMPO)  GOTO 3420
000   C              THEN
000   C                  BY PASS THIS RECORD
000                      TBYPAS = TBYPAS + 1
000               IF  (MOD(COUNT,50) .EQ. 0) CALL  PRTFAS (COUNT,4,FASTAB)
000                      GOTO 3440
000   C              ELSE
000   C                  PROCESS ONE EACH LOCKOUT REC WITH CURRENT FAS REC
000    3420              TLKOUT = TLKOUT + 1
000                      ROBCO = LOTAB(2)
000                      AUTH8  = X
000                      AUTH13 = BLANK
000                      AUTH47 = BLANK
000                      AUTH9A = BLANK
000                      AUTHBD = BLANK
000                      AUTHE  = BLANK
000               IF  (MOD(COUNT,50) .EQ. 0) CALL  PRTFAS (COUNT,5,FASTAB)
000                      ENCODE (312,1,FASREC)  FASTAB
000                      WRITE (FILEF)  FASREC
000   C            END-IF
000    3440        CONTINUE
000   C       END-IF
000           NEXT = FAS
000           READ (LOFILE,2,END=3499)  LOTAB
000           GOTO 3970
000    3499   LOUIC = -1
000           GOTO 3970
000   C
000   C  •••  COMPARE  HAND-PLAYED (UIC)  WITH  FAS (UIC)  •••
000   C
000    3500   IF  (HPUIC .NE. FASUIC)  GOTO 3600
000   C         THEN
000   C            PROCESS ONE EACH HAND-PLAYED UNIT WITH CURRENT FAS REC
000                THP = THP + 1
000                ROBCO = HPTAB(3)
000                AUTH8 = ONE
000                AUTHBD = HPTAB(6)
000                DECODE (3,4,HPTAB(6))  ROD
000        4  FORMAT (I3)
000        5  FORMAT (J2)
```

```
000                  CALL   GETTP (RDD,TIMEPD)
000                  ENCODE (2,5,AUTH9A)  TIMEPD
000                  ENCODE (2,102,AUTHE) BLANK, ONE
000                  AUTH13 = BLANK
000                  AUTH47 = BLANK
000                 IF  (MOD(COUNT,50) .EQ. 0) CALL  PRTFAS (COUNT,6,FASTAB)
000                  ENCODE (312,1,FASREC)  FASTAB
000                  WRITE (FILEF)  FASREC
000                  NEXT = FAS
000                  READ (HPFILE,8,END=3599)  HPTAB
000   C         END-IF
000             GOTO 3960
000    3599     HPUIC = -1
000             GOTO 3960
000   C
000   C  ***  CHECK  FOR NOTIONAL FROM MRFILE  ***
000   C
000    3600     IF  (MUIC .EQ. FASUIC)  GOTO 3650
000   C           THEN
000   C              PROCESS  ONE EACH NOTIONAL UNIT
000                   IF (MRFLAG .NE. '0')  GOTO 3995
000                  TSHORT = TSHORT + 1
000                  NEXT = NOT
000                  MRTAB(29) = BLANK
000                 IF  (MOD(COUNT,50) .EQ. 0) CALL  PRTFAS (COUNT,7,NOTTAB)
000                  I = 1
000   C
000   C              DOWHILE     (I .LE. 44)
000   C
000    3620                      IF  (I .GT. 44)  GOTO 3640
000                          K = NINDEX(I)
000                          NOTTAB(K) = MRTAB(I)
000                          I = I + 1
000                          GOTO 3620
000   C              END-DO
000   C
000    3640          ENCODE (2,102,NOTTAB(14)) NOTTAB(14), ONE
000             NOTTAB(82) = BLANK
000                  ENCODE (312,1,NOTREC)  NOTTAB
000                  WRITE (FILEF)  NOTREC
000                  GOTO 3700
000   C           ELSE
000   C              THIS UNIT IS EITHER A MATCH-REQ UNIT OR AN OVERAGE
000    3650          IF  (MRFLAG .NE. '1')  GOTO 3660
000   C                 THEN
000   C                  PROCESS ONE EACH MATCH-REQ WITH CURRENT FAS-REC
000                          TMATCH = TMATCH + 1
000                          ROBCO = MRTAB(22)
000                          AUTH13 = MRTAB(7)
000                          AUTH47 = MRTAB(8)
000                          AUTH8  = MRTAB(9)
000                          AUTH9A = MRTAB(10)
000                          AUTHBD = MRTAB(11)
000                          ENCODE (2,102,AUTHE) MRTAB(12), ONE
000                 IF  (MOD(COUNT,50) .EQ. 0) CALL  PRTFAS (COUNT,8,FASTAB)
000                          GOTO 3680
000   C                 ELSE
```

```
000   C                  PROCESS ONE EACH OVERAGE WITH CURRENT FAS REC
000    3660              TOVER = TOVER + 1
000                      ROBCO = MRTAB(22)
000                      AUTH13 = ZERO
000                      AUTH47 = ZERO
000                      AUTH8  = ZERO
000                      AUTH9A = NINES
000                      AUTHBD = NINES
000                      AUTHE  = ZERO
000                IF  (MOD(COUNT,50) .EQ. 0) CALL  PRTFAS (COUNT,9,FASTAB)
000   C             END-IF
000    3680         NEXT = FAS
000                 ENCODE (312,1,FASREC)  FASTAB
000                 WRITE (FILEF)  FASREC
000   C        END-IF
000    3700    CONTINUE
000            CALL  SRRET (MRREC,DUMMY,$3799)
000            DECODE (150,3,MRREC)  MRTAB
000            GOTO 3950
000    3799    MUIC = -1
000   C
000   C
000    3950    CONTINUE
000   C
000   C
000    3960    CONTINUE
000   C
000   C
000    3970    CONTINUE
000   C
000   C
000    3980    CONTINUE
000   C
000   C
000    3990    CONTINUE
000            GOTO 3000
000   C  END-DO
000   C
000    3995    WRITE (6,99)  LUIC, FASUIC
000      99    FORMAT ('  *** LUIC=',A6,' FASUIC = ',A6,', WAS NOT FOUND ***')
000            CALL  PRTFAS(-1,9,FASTAB)
000    3999    CONTINUE
000            TOTAL = TALT+TATLOV+TATL+TBYPAS+TLKOUT+THP+TSHORT+TMATCH+TOVER
000            WRITE (6,101) TALT,TATLOV,TATL,TBYPAS,TLKOUT,THP,TSHORT,
000        *                 TMATCH,TOVER,TOTAL
000     101    FORMAT(' TALT=',I6,' TATLOV=',I6,' TATL=',I6,' TBYPAS=',I6,
000        *         ' TLKOUT=',I6//' THP=',I6,' TSHORT=',I6,' TMATCH=',I6,
000        *         ' TOVER=',I6,' TOTAL =',I8//' *** LAYIN FINI ***')
000     102    FORMAT (2A1)
000       8    FORMAT (9X,A1,6X,A6,33X,A3,17X,A1,A2,A3,6X,A6,A3)
000            ENDFILE  11
000            ENDFILE  12
000            ENDFILE  13
000            ENDFILE  14
000            ENDFILE  15
000            ENDFILE  16
000            ENDFILE  17
```

```
000          STATUS =  CSF  (2,'@BRKPT  18 ')
000          STOP
000          END

END ELT.


@ELT,L   FSLAY.PRTFAS
ELT007 RLIB70 08/16-07:40:43-(4,)
001      C
001      C                      P   R   T   F   A   S
001      C
004          SUBROUTINE  PRTFAS (I,J,FASTAB)
001      C
001          IMPLICIT  INTEGER  (A - Z)
004          INTEGER   FASTAB(99)
001          INTEGER  REASON(9)
001          DATA  REASON/'ALT-TH','ATL-OV','ATL   ','LO-BYP',
001         •     'LCKOUT','HPUNIT','NOTION','MATCH','OVERAG'/
001      C
001          WRITE (18,1)  REASON(J), (FASTAB(K),K=1,34)
001          WRITE (18,2)             (FASTAB(K),K=35,70)
001          WRITE (18,3)             (FASTAB(K),K=71,82)
001          WRITE (18,4)          I, (FASTAB(K),K=83,99)
001      C
001          RETURN
001      C
003        1 FORMAT (1H0,A6,1X,2A1,1X,2(A6,1X),A1,1X,A6,1X,A4,1X,A2,1X,
001         • A3,1X,A4,1X,A1,1X,A2,1X,A3,1X,3(A2,1X),A6,1X,2(A2,1X),A5,1X,
002         • A3,1X,A2,1X,A3,1X,2(A2,1X),5(A1,1X),A3,1X,A5,1X,A3,1X,A3)
002        2 FORMAT (1H ,3X,A2,1X,A3,1X,A2,1X,A6,1X,A3,1X,A2,1X,A1,1X,A2,1X,
001         •   A1,1X,A3,1X,A1,1X,A2,1X,A3,1X,5(A1,1X),A3,1X,A1,1X,A5,1X,
001         •   2(A2,1X),4(A1,1X),A5,1X,A2,1X,A6,1X,A3,1X,A5,1X,A6,1X,A5,1X,
001         •   A1,1X,A5)
002        3 FORMAT (1H ,3X,4(A1,1X),A5,1X,A4,1X,A6,1X,A3,1X,3A6,1X,A3)
001        4 FORMAT (1H ,16,2X,15(A5,1X),A4,1X,A1)
001          END

END ELT.


@HDG     FMA POST-PROCESSOR PROGRAM BINCOPY        ••UNCLASSIFIED•• .L,0
```

```
@ELT,L   24MONTAGNE.BINCOPY-SUM
ELT007 RLIB70 08/16-07:40:45-(10,)
006            IDENTIFICATION DIVISION.
006            PROGRAM-ID.  BINCOPY-A.
006            AUTHOR.   E MONTAGNE.
006            DATE WRITTEN.      17 JULY 1975.
006            REMARKS.  THIS PROGRAM IS PART OF A SERIES OF NEW CAMP
006                PROGRAMS.  IT CALLS A FORTRAN SUBROUTINE TO READ AN
006             UNFORMATTED BINARY FILE (BIFAS) INTO A COBOL FILE CALLED FASBIN307.
006                BIFAS IS CREATED BY KELLER TO CONTAIN RESULTS OF A MATCH OF
006                CONWAY'S OUTPUT AGAINST THE FASTALS OUTPUT.
006                FASBIN307 IS THE FAS FILE (307 CHARACTERS PER RECORD, 24 RECORDS
006               PER BLOCK) H WHICH IS THEN SORTED AND OUTPUT TO TAPE (FAS-OUT FILE).
006                           SELECTED FIELDS ARE LISTED TO
006                ASSURE CORRECTNESS PRIOR TO DELIVERY OF THETAPE.
006                REVISED 24 JULY TO REPLACE BLANKS WITH ZERO
006                IN SELECTED FIELDS (AUTHR ESCON EQCON).
006       REVISED 15 AUG FOR TOTAL FORCE.
006
006            ENVIRONMENT DIVISION.
006            CONFIGURATION SECTION.
006            SOURCE-COMPUTER.  UNIVAC-1108.
006            OBJECT-COMPUTER.  UNIVAC-1108.
006            INPUT-OUTPUT SECTION.
006            FILE-CONTROL.
006                SELECT PRINT-FILE ASSIGN TO PRINTER.
006                SELECT SORT-ING ASSIGN TO MASS-STORAGE XA.
006                SELECT FAS-OUT ASSIGN TO UNISERVO TAPE-FILE.
006                SELECT ALT-PRINT-FILE ASSIGN TO PRINTER 24FAS-PRINT.
006            DATA DIVISION.
006            FILE SECTION.
006            SD  SORT-ING.
006            01  SORT-REC.
006                02  FILLER     PICTURE  XX.
006                02  COMPO        PICTURE 9.
006                02  UICCC          PICTURE  X(6).
006                02  EDATE             PICTURE    9(6).
006                02  FILLER          PICTURE X(13).
006                02  AUTHR  PICTURE X(15).
006                02  FILLER PICTURE X(31).
006                02  EQCON  PICTURE X.
006                02  ESCON  PICTURE X.
006                02  FILLER PICTURE X(231)
006            FD  ALT-PRINT-FILE      LABEL RECORD OMITTED.
006            01  ALT-REC           PIC X(132).
006            FD  FAS-OUT BLOCK 24 RECORDS LABEL RECORD OMITTED RECORDING
006                MODE IS 1.
006            01  FAS-REC.
006                02  FILLER          PIC X.
006                02  FICOD         PIC X.
006                02  COMPO            PIC  X.
006                02  UICCC            PIC  X(6).
006                02  EDATE            PIC  9(6).
006                02  FILLER           PIC  X(13).
006                02  AUTHR.
006                    03  UIN  PIC XXX.
006                    03  FLAGS  PIC XXXX.
```

```
006                        03  MATCH-CODE    PIC X.
006                        03  TP    PIC XX.
006                        03  RDD   PIC XXX.
006                        03  LRE          PIC X.
006                        03  TH-CODE           PICTURE X.
006                    02  BRNCH           PIC  XX.
006                    02  CARSS           PIC  XX.
006                    02  FILLER          PIC  X(23).
006                    02  DSCMP           PIC  XX.
006                    02  FILLER          PIC  XX.
006                    02  EQCON           PIC  X.
006                    02  ESCON           PIC  X.
006                    02  FILLER          PIC  XXX.
006                    02  FPLAN     PIC XXX.
006                    02  JCSTY           PIC  X(5).
006                    02  LOCCO        PIC X(3).
006                    02  FILLER       PIC X(39).
006                    02  ROBCO           PIC  XXX.
006                    02  FILLER          PIC  X.
006                    02  SRCTO           PIC  X(13).
006                    02  FILLER          PIC  X(7).
006                    02  STNNM           PIC  X(9).
006                    02  FILLER          PIC  X(17).
006                    02  TPSNA           PIC  X(5).
006                    02  FILLER          PIC  X(9).
006                    02  UNMBR           PIC  X(4).
006                    02  PECOD           PIC  X(6).
006                    02  ULCCC           PIC  X(3).
006                    02  UNTDS-18        PIC  X(18).
006                    02  FILLER          PIC  X(3).
006                    02  AUOFF    PIC X(5).
006                    02  AUWOF    PIC X(5).
006                    02  AUENL    PIC X(5).
006                    02  AUAGR    PIC X(5).
006                    02  AUCIV    PIC X(5).
006                    02  FILLER   PIC X(15).
006                    02  OPAGR    PIC X(5).
006                    02  FILLER   PIC X(5).
006                    02  STOFF    PIC X(5).
006                    02  STWOF    PIC X(5).
006                    02  STENL    PIC X(5).
006                    02  STAGR    PIC 9(5).
006                    02  STCIV    PIC X(5).
006                    02  FILLER   PIC X(5).
006                FD  PRINT-FILE LABEL RECORD OMITTED.
006                01  LINE-REC-3  PIC X(132).
006                WORKING-STORAGE SECTION.
006                77  K     PIC 99  USAGE IS COMPUTATIONAL.
006                77  J     PIC 99  USAGE IS COMPUTATIONAL.
006                77  I     PIC 99  USAGE IS COMPUTATIONAL.
006                77  TIME-PERIOD    PIC  99.
006                77  COMPO-CODE    PIC X.
006                77  COMPO-CODE-1    PIC  9.
006                01  STAGR-TABLE.
006                    02  STAGR-TP    OCCURS  10 TIMES.
006                      03  STAGR-SUM OCCURS 9 TIMES  PIC 999999.
006                01  SUM-TABLE.
```

```
006           02  CUM-STAGR-TP    OCCURS 10 TIMES.
006           03  CUM-STAGR-SUM OCCURS 9 TIMES PIC 9(7).
006       77  COUNT-REC  PICTURE 9(5) USAGE IS COMPUTATIONAL VALUE ZERO.
006       77  LINE-NR          PIC  99   USAGE IS COMPUTATIONAL.
006       77  IEND-OF-FILE     PICTURE 9  VALUE 2.
006       77  FASBIN-REC-312     PICTURE x(312).
006       01  LINE-REC       VALUE SPACES.
006           02  SEQNR              PIC 9(5).
006           02  FILLER             PIC X.
006           02  COMPO              PIC X.
006           02  FILLER             PIC X(5).
006           02  UICCC              PIC X(6).
006           02  FILLER             PIC X.
006           02  EDATE              PIC 9(6).
006           02  FILLER             PIC X.
006           02  SRCTO              PIC X(13).
006           02  FILLER             PIC X.
006           02  TPSNA              PIC X(5).
006           02  FILLER             PIC X.
006           02  FPLAN      PIC x(3).
006           02  FILLER     PIC x(5).
006           02  CARSS              PIC XX.
006           02  FILLER             PIC X.
006           02  UNMBR              PIC X(4).
006           02  FILLER             PIC XX.
006           02  BRNCH              PIC X(2).
006           02  FILLER             PIC X.
006           02  ULCCC              PIC X(3).
006           02  FILLER             PIC X.
006           02  UNTDS-18           PIC X(18).
006           02  FILLER             PIC X.
006           02  STNNM              PIC X(9).
006           02  FILLER             PIC X.
006           02  LOCCO              PIC X(3).
006           02  FILLER             PIC XXX.
006           02  AUTHR.
006               03  UIN      PIC XXX.
006               03  FILLER   PIC X.
006               03  FLAGS      PIC XXXX.
006               03  FILLER   PIC X.
006               03  MATCH-CODE     PIC X.
006               03  FILLER   PIC X.
006               03  TP       PIC XX.
006               03  FILLER   PIC X.
006               03  RDD      PIC XXX.
006               03  FILLER   PIC X.
006               03  LRE      PIC X.
006               03  FILLER   PIC X.
006               03  TH-CODE              PIC X.
006           02  FILLER     PIC X.
006           02  FILLER             PIC XX.
006           02 FILLER        PIC XXX.
006       01  LINE-REC-2       VALUE SPACES.
006           02  FILLER    PIC    X(11).
006           02  DSCMP    PIC XX.
006           02  FILLER   PIC X(4).
006           02  JCSTY              PIC X(5).
```

```
006              02  FILLER          PIC X.
006              02  ROBCO           PIC XXX.
006              02  FILLER          PIC XXX.
006              02  PECOD    PIC X(6).
006              02  FILLER   PIC X.
006              02  STAGR    PIC X(5).
006              02  FILLER   PIC X(91).
006          01  FASBIN-REC-307.
006              02  FILLER      PICTURE  X.
006             02  FICODE     PICTURE   X.
006              02  COMPO  PICTURE 9.
006              02  UICCC        PICTURE  X(6).
006              02  EDATE          PICTURE   9(6).
006              02  FILLER        PICTURE X(13).
006              02  AUTHR  PIC X(15).
006              02  FILLER PICTURE X(31).
006              02  EQCON  PICTURE X.
006              02  ESCON  PICTURE X.
006              02  FILLER PICTURE X(151).
006              02  STRENGTHS  PIC  X(75).
006              02  FILLER      PIC  X(5).
006          01  HEADER-1.
006              02  FILLER PIC X(89) VALUE 'SEQNR COMPO UICCC  EDATE  SRCTO -
006         -    ' - - - TPSNA FPLAN   CA UNMBR BR ULC UNTDS  - - - - - - '.
006              02  FILLER     PIC X(35) VALUE 'STNNM     LOCCO AUTHR - - - -
006         -    '- - -'.
006              02  FILLER     PIC X(8)  VALUE SPACES.
006          01  HEADER-2.
006              02  FILLER     PIC X(44)  VALUE '          DSC   JCSTY ROBCO
006         -    'PECOD STAGR'.
006              02  FILLER          PIC X(88)  VALUE SPACES.
006          PROCEDURE DIVISION.
006          OPEN-UP SECTION.
007              OPEN OUTPUT FAS-OUT   PRINT-FILE ALT-PRINT-FILE.
006          FAS-SORT SECTION.
006              SORT SORT-ING ON ASCENDING KEY COMPO UICCC EDATE
006              INPUT PROCEDURE IS ENTER-FORTRAN
006              OUTPUT PROCEDURE IS WRITE-TAPE.
006              GO TO END-RUN.
006          ENTER-FORTRAN SECTION.
006              ENTER FORTRAN READBI SUBROUTINE REFERENCING
006                   FASBIN-REC-312  IEND-OF-FILE.
006              IF IEND-OF-FILE = 4  GO TO CLOSE-307.
006              ADD 1 TO COUNT-REC.
006              MOVE FASBIN-REC-312 TO FASBIN-REC-307.
006              EXAMINE STRENGTHS REPLACING ALL SPACES BY ZERO.
006               RELEASE SORT-REC FROM FASBIN-REC-307.
006              GO TO ENTER-FORTRAN.
006          CLOSE-307.
006              MOVE SPACES TO LINE-REC MOVE COUNT-REC TO SEQNR.
006                      MOVE 'RECORDS      ' TO SRCTO IN LINE-REC.
006              WRITE LINE-REC-3 FROM LINE-REC.
006              MOVE ZERO TO SEQNR.
006          WRITE-TAPE SECTION.
008                PERFORM SET-ZERO VARYING J FROM 1 BY 1 UNTIL J = 11.
006          SET-HEADER.
006              MOVE ZERO TO LINE-NR.
```

```
007              WRITE ALT-REC FROM HEADER-1 AFTER ADVANCING TOP-OF-PAGE.
007              WRITE ALT-REC FROM HEADER-2  MOVE SPACES TO LINE-REC-3.
007              WRITE ALT-REC.
006          WRITE-TAPE-LINES.
006              RETURN SORT-ING RECORD INTO FAS-REC AT END GO TO PARA-1.
009               IF TP IN FAS-REC NOT NUMERIC OR COMPO IN FAS-REC
009                      NOT NUMERIC GO TO SKIP-COMP.
006              MOVE TP IN FAS-REC TO TIME-PERIOD.
009              IF TIME-PERIOD < 1 OR > 8 GO TO SKIP-COMP.
009              MOVE COMPO IN FAS-REC TO COMPO-CODE-1.
006              ADD STAGR IN FAS-REC TO
006                 STAGR-SUM(TIME-PERIOD, COMPO-CODE-1).
009          SKIP-COMP.
006              ADD 1 TO SEQNR.
006              MOVE CORRESPONDING FAS-REC TO LINE-REC.
007                WRITE ALT-REC FROM   LINE-REC.
006              MOVE CORRESPONDING FAS-REC TO LINE-REC-2.
007              WRITE ALT-REC FROM LINE-REC-2  WRITE FAS-REC.
006              ADD 2 TO LINE-NR.
006              IF LINE-NR = 54 GO TO SET-HEADER.
006              GO TO WRITE-TAPE-LINES.
006          PARA-1.
006              EXIT.
006          END-RUN SECTION.
006              PERFORM COMPUTE-SUM VARYING I FROM 1 BY 1 UNTIL
006                      I = 11.
006              PERFORM COMP-CUM-SUM VARYING J FROM 2 BY 1
006                      UNTIL J  = 11.
006              CLOSE PRINT-FILE.
007              CLOSE FAS-OUT       ALT-PRINT-FILE.
006              STOP RUN.
006          SET-ZERO.
006              PERFORM SET-1 VARYING I FROM 1 BY 1 UNTIL I = 10.
006          SET-1.
006              COMPUTE  STAGR-SUM(J, I) = 0.
006              COMPUTE  CUM-STAGR-SUM(J, I) = 0.
006          COMPUTE-SUM.
009              COMPUTE STAGR-SUM(I, 6) = STAGR-SUM(I, 2) +
009                        STAGR-SUM(I, 3).
009              COMPUTE STAGR-SUM(I, 7) = STAGR-SUM(I, 6) +
009                        STAGR-SUM(I, 4).
009              COMPUTE STAGR-SUM(I, 8) = STAGR-SUM(I, 1) +
006                        STAGR-SUM(I, 6).
009              COMPUTE STAGR-SUM(I, 9) = STAGR-SUM(I, 7) +
009                        STAGR-SUM(I, I).
006              MOVE STAGR-TP(I) TO LINE-REC-3  WRITE LINE-REC-3.
006          COMP-CUM-SUM.
006              PERFORM COMP-CUM-SUM-1 VARYING I FROM 1 BY 1 UNTIL
006                        I = 10.
006             MOVE CUM-STAGR-TP(J) TO LINE-REC-3  WRITE
006                        LINE-REC-3.
006          COMP-CUM-SUM-1.
006             COMPUTE K = J - 1  COMPUTE CUM-STAGR-SUM(J, I) =
010                       CUM-STAGR-SUM(K, I) + STAGR-SUM(K, I).

END ELT.
```

```
@ELT,L   24MONTAGNE.READBIN
ELT007 RLIB70 08/16-07:40:48-(29,)
      025           SUBROUTINE READBI (FASBIN,IEND)
      025            DATA IFILE/2/ICOUNT/0/KCOUNT/0/LCOUNT/0/
      025           DIMENSION FASBIN(52)
      029    C          THIS SUBROUTINE READS A BINARY UNFORMATTED FORTRAN FILE.
      029    C          THE FILE CONTAINS FAS RECORDS WITH MATCH RESULTS.
      025    C       RECORDS ARE 52 WORDS (312 CHARACTERS).  READBIN IS MEANT TO
      025    C        BE CALLED BY A COBOL PROGRAM.
      025    C      DO STATEMENT FOR DEBUG ONLY.
      025    C       DO 200 I = 1, 50
      029    200      READ (IFILE,END=30) FASBIN
      025            KCOUNT = KCOUNT + 1
      025    120     RETURN
      025    199     FORMAT (10X,I5,5X,I5)
      025    30      IEND = 1H4
      025           PRINT 199, KCOUNT
      025           RETURN
      025           END

END ELT.

@HDG     MRG  PRE-PROCESSOR PROGRAM SORTARLOC      ••UNCLASSIFIED•• •L,D
```

```
@ELT,L    24MONTAGNE.ARLOCSORT
ELT007 RLIB70 08/16-07:40:49-(1,)
       000         IDENTIFICATION DIVISION.
       000         PROGRAM-ID.  ARLOCSORT.
       000         AUTHOR.     E MONTAGNE.
       000         DATE-WRITTEN.    9 JULY 1975.
       000         REMARKS.  INPUT IS ARLOC TAPE FROM AOC.  TAPE CONTENTS ARE
       000             PRINTED, THEN SORTED BY LOCNA, AREA, AND TLAC.  THE SORTED
       000             FILE  IS PRINTED AND OUTPUT TO TAPE.
       000         ENVIRONMENT DIVISION.
       000         CONFIGURATION SECTION.
       000         SOURCE-COMPUTER.  UNIVAC-1108.
       000         OBJECT-COMPUTER.  UNIVAC-1108.
       000         INPUT-OUTPUT SECTION.
       000         FILE-CONTROL.
       000             SELECT AFTER-SORT ASSIGN TO UNISERVO TAPE-SORT.
       000             SELECT SORT-ING ASSIGN TO MASS-STORAGE XA.
       000             SELECT PRINT-FILE ASSIGN TO PRINTER.
       000             SELECT TAPE-FILE ASSIGN TO UNISERVO TAPE-FILE.
       000         DATA DIVISION.
       000         FILE SECTION.
       000         SD  SORT-ING.
       000         01  SORTGEOREC.
       000             02  AREA            PICTURE X(3).
       000             02  FILLER          PICTURE X(5).
       000             02  ARLOC           PICTURE X(5).
       000             02  FILLER          PICTURE X(4).
       000             02  GELOC           PICTURE X(4).
       000             02  FILLER          PICTURE X(5).
       000             02  LOCNA           PICTURE X(9).
       000             02  FILLER          PICTURE X(4).
       000             02  LOCNM           PICTURE X(17).
       000             02  FILLER          PICTURE X(8).
       000             02  TLAC            PICTURE X(3).
       000             02  FILLER          PICTURE X(10).
       000             02  FCTCD           PICTURE X(18).
       000             02  FILLER          PICTURE X(2).
       000             02  UGRID           PICTURE X(11).
       000             02  FILLER          PICTURE X(3).
       000             02  LFLLC           PICTURE X(15).
       000             02  FILLER          PICTURE X(2).
       000             02  STAT            PICTURE X(1).
       000             02  FILLER          PICTURE X(3).
       000         FD  AFTER-SORT LABEL RECORD OMITTED.
       000         01  SORTED-REC          PICTURE X(132).
       000         FD  PRINT-FILE LABEL RECORD OMITTED.
       000         01  PRINT-LINE PICTURE X(132).
       000         FD  TAPE-FILE LABEL RECORD OMITTED RECORDING MODE IS 1.
       000         01  TAPE-REC     PICTURE X(132).
       000         WORKING-STORAGE SECTION.
       000         01  HEADER-LINE.
       000             02  FILLER PICTURE X(26) VALUE 'AREA     ARLOC     GELOC     '.
       000             02  FILLER PICTURE X(26) VALUE 'LOCNA          LOCNM        '.
       000             02  FILLER PICTURE X(25) VALUE '              TLAC         '.
       000             02  FILLER PICTURE X(25) VALUE 'FCTCD                  UGRID'.
       000             02  FILLER PICTURE X(26) VALUE '           LFLLC            '.
       000             02  FILLER PICTURE X(4)  VALUE 'STAT'.
```

```
000    01  TOP-HEADER.
000        02  FILLER PICTURE X(55) VALUE SPACES.
000        02  FILLER PICTURE X(23) VALUE 'DA GEOGRAPHIC LOCATIONS'.
000        02  FILLER PICTURE X(41) VALUE SPACES.
000        02  FILLER PICTURE X(5)  VALUE 'PAGE'.
000        02  PAGENR PICTURE 999   VALUE 0.
000        02  FILLER PICTURE X(5)  VALUE SPACES.
000        01  LAST-LINE.
000        02  FILLER PICTURE X(24) VALUE '  NUMBER OF RECORDS IS  '.
000        02  O-REC-COUNT PICTURE ZZZZZ.
000        02  FILLER PICTURE X(103) VALUE SPACES.
000    77  LINE-COUNT PICTURE 99.
000    77  REC-COUNT  PICTURE 99999 VALUE ZERO.
000    PROCEDURE DIVISION.
000    OPEN-FILES.
000        OPEN INPUT TAPE-FILE.
000        OPEN OUTPUT PRINT-FILE.
000        OPEN OUTPUT AFTER-SORT.
000    A-SORT SECTION.
000        SORT SORT-ING ON ASCENDING KEY LOCNA AREA TLAC
000        INPUT PROCEDURE IS I-SORT
000        OUTPUT PROCEDURE IS O-SORT.
000        GO TO PRINT-SORTED.
000    I-SORT SECTION.
000    SET-HEADER.
000        ADD 1 TO PAGENR  MOVE 1 TO LINE-COUNT.
000        WRITE PRINT-LINE FROM TOP-HEADER AFTER ADVANCING
000              TOP-OF-PAGE.
000        MOVE SPACES TO PRINT-LINE  WRITE PRINT-LINE.
000        WRITE PRINT-LINE FROM HEADER-LINE.
000        MOVE SPACES TO PRINT-LINE  WRITE PRINT-LINE.
000    READ-FILE.
000        READ TAPE-FILE AT END GO TO END-I-SORT.
000        ADD 1 TO REC-COUNT  WRITE PRINT-LINE FROM TAPE-REC.
000        RELEASE SORTGEOREC FROM TAPE-REC.
000        IF LINE-COUNT EQUAL TO 53 GO TO SET-HEADER.
000        ADD 1 TO LINE-COUNT  GO TO READ-FILE.
000    END-I-SORT.
000        CLOSE TAPE-FILE MOVE REC-COUNT TO O-REC-COUNT.
000        WRITE PRINT-LINE FROM LAST-LINE.
000    O-SORT SECTION.
000        RETURN SORT-ING RECORD INTO SORTED-REC  AT END GO TO
000            END-O-SORT.
000        WRITE SORTED-REC  GO TO O-SORT.
000    END-O-SORT.
000        CLOSE AFTER-SORT.
000    PRINT-SORTED SECTION.
000        OPEN INPUT AFTER-SORT.
000      MOVE 0 TO PAGENR.
001        MOVE 0 TO REC-COUNT.
000    RESET-HEADER.
000        ADD 1 TO PAGENR  MOVE 1 TO LINE-COUNT.
000        WRITE PRINT-LINE FROM TOP-HEADER AFTER ADVANCING
000            TOP-OF-PAGE.
000        WRITE PRINT-LINE FROM HEADER-LINE AFTER 2.
000        MOVE SPACES TO PRINT-LINE  WRITE PRINT-LINE.
000    READ-SORTED-FILE.
```

```
000          READ AFTER-SORT AT END GO TO END-RUN.
000          ADD 1 TO REC-COUNT  WRITE PRINT-LINE FROM SORTED-REC.
000          IF LINE-COUNT EQUAL TO 53 GO TO RESET-HEADER.
000          ADD 1 TO LINE-COUNT  GO TO READ-SORTED-FILE.
000      END-RUN.
000          MOVE REC-COUNT TO O-REC-COUNT.
000          WRITE PRINT-LINE FROM LAST-LINE.
000          CLOSE AFTER-SORT  PRINT-FILE.
000          STOP RUN.

END ELT.


@ELT.L   24MONTAGNE.ARLOC-23
ELT007 RLIB70 08/16-07:40:51-(14,)
010      IDENTIFICATION DIVISION.
010      PROGRAM-ID.  ARLOC-23.
010      AUTHOR.  E MONTAGNE.
010      DATE-WRITTEN.  12 SEPT 75.
010      REMARKS.  PROGRAM SORTS SELECTED FIELDS OF ARLOC FILE AND
010                PUTS RESULTS IN FILE 24SORTARLOC.
010      ENVIRONMENT DIVISION.
010      CONFIGURATION SECTION.
010      SOURCE-COMPUTER.  UNIVAC-1108.
010      OBJECT-COMPUTER.  UNIVAC-1108.
010      INPUT-OUTPUT SECTION.
010      FILE-CONTROL.
010          SELECT OUT-FILE ASSIGN TO PRINTER 24SORTARLOC.
010          SELECT SORT-ING ASSIGN TO MASS-STORAGE XA.
010          SELECT PRINT-FILE ASSIGN TO PRINTER.
010          SELECT ARLOC-FILE ASSIGN TO UNISERVO ARLOC-FILE.
010      DATA DIVISION.
010      FILE SECTION.
010      SD  SORT-ING.
010      01  SORTGEOREC.
011          02  AREA-AREA       PIC X(3).
010          02  LOCNA      PIC X(9).
010          02  GELOC      PIC X(4).
010          02  ARLOC      PIC X(5).
014          02  SAVE-AREA-1     PIC X.
010          02  TLAC       PIC X(3).
010      FD  ARLOC-FILE LABEL RECORD  OMITTED RECORDING MODE IS 1.
010      01  ARLOC-REC.
011          02  AREA-AREA       PIC X(3).
010          02  FILLER      PIC X(5).
010          02  ARLOC       PIC X(5).
010          02  FILLER      PIC X(4).
010          02  GELOC       PIC X(4).
010          02  FILLER      PIC X(5).
010          02  LOCNA       PIC X(9).
010          02  FILLER      PIC X(4).
010          02  LOCNM       PIC X(17).
010          02  FILLER      PIC X(8).
010          02  TLAC        PIC X(3).
010          02  FILLER      PIC X(10).
010          02  FCTCD       PIC X(18).
```

```
010          02  FILLER      PIC X(2).
010          02  UGRID       PIC X(11).
010          02  FILLER      PIC X(3).
010          02  LFLLC       PIC X(15).
010          02  FILLER      PIC X(2).
010          02  STAT        PIC X(1).
010          02  FILLER      PIC X(3).
010      FD  OUT-FILE  LABEL RECORD OMITTED.
010      01  OUT-REC.
011          02  AREA-AREA        PIC X(3).
010          02  LOCNA       PIC X(9).
010          02  GELOC       PIC X(4).
010          02  ARLOC       PIC X(5).
011          02  SAVE-AREA-1     PIC X.
010       FD  PRINT-FILE LABEL RECORD OMITTED.
010       01  PRINT-LINE      PIC X(132).
010       WORKING-STORAGE SECTION.
010       01  SORT-REC.
011           02  AREA-AREA.
011               03  AREA-1   PIC X.
011               03  AREA2-3  PIC XX.
010           02  LOCNA     PIC X(9).
010           02  GELOC     PIC X(4).
010           02  ARLOC     PIC X(5).
014           02  SAVE-AREA-1     PIC X.
014           02  TLAC       PIC X(3).
010       01  LAST-LINE.
010           02  FILLER  PIC XX  VALUE SPACES.
010           02  O-REC-COUNT   PIC ZZZZZ.
010           02  FILLER        PIC X(9)  VALUE IS '  RECORDS'.
010           02  FILLER        PIC X(116)   VALUE SPACES.
010      01  TEMP-TABLE.
010          02  TEMP  OCCURS 6 TIMES.
011              03  AREA-AREA    PIC X(3).
010              03  LOCNAME   PIC X(9).
010              03  GELOC     PIC X(4).
010              03  ARLOC     PIC X(5).
012              03  SAVE-AREA-1    PIC X.
014              03  TLACE     PIC X(3).
010      77  KAIN   PIC 9 USAGE IS COMPUTATIONAL VALUE  ZERO.
010      77  KCTY   PIC 9 USAGE IS COMPUTATIONAL VALUE  ZERO.
010      77  I      PIC 99 USAGE IS COMPUTATIONAL VALUE  ZERO.
010      77  J      PIC 99 USAGE IS COMPUTATIONAL.
010       77  REC-COUNT PIC 99999 USAGE IS COMPUTATIONAL VALUE ZERO.
010       PROCEDURE DIVISION.
010           OPEN INPUT ARLOC-FILE OPEN OUTPUT OUT-FILE PRINT-FILE.
010       A-SORT SECTION.
011           SORT SORT-ING ON ASCENDING KEY AREA-AREA LOCNA
010           INPUT PROCEDURE IS I-SORT
010           OUTPUT PROCEDURE IS O-SORT.
010           GO TO END-RUN.
010       I-SORT SECTION.
010       READ-FILE.
010           READ ARLOC-FILE AT END GO TO END-I-SORT.
010           ADD 1 TO REC-COUNT.
010           MOVE CORRESPONDING ARLOC-REC TO SORT-REC.
012         IF AREA-1 IN SORT-REC IS NUMERIC PERFORM
```

```
012                SCRATCH-NUMERIC.
010           RELEASE SORTGEOREC FROM SORT-REC.
010           GO TO READ-FILE.
012       SCRATCH-NUMERIC.
013           MOVE AREA-1 IN SORT-REC TO SAVE-AREA-1 IN SORT-REC.
012           MOVE SPACE TO AREA-1 IN SORT-REC.
010       END-I-SORT.
010           MOVE REC-COUNT TO O-REC-COUNT.
010           WRITE PRINT-LINE FROM LAST-LINE.
010           MOVE  ZERO TO REC-COUNT MOVE SPACES TO PRINT-LINE.
010       O-SORT SECTION.
010       *
010       *      ELIMINATE DUPLICATE LOCNA RECORDS.              *
010       *
010       CHECK-DUP.
010           ADD 1 TO I.
010           RETURN SORT-ING RECORD INTO TEMP(I) AT END GO TO END-O-SORT.
010           IF LOCNAME(I) = LOCNAME(1) GO TO CHECK-DUP.
010           PERFORM SELECT-LOCNA THRU LAST-SEL.
010           MOVE TEMP(I) TO TEMP(1)  MOVE 1 TO I  GO TO CHECK-DUP.
010       *
010       *   SELECT LOCNA = AIN OR CTY IF POSSIBLE,
010       *   ELSE ARBITRARILY SELECT TEMP(1).
010       *
010       SELECT-LOCNA.
010           PERFORM SELECT-AIN-CTY VARYING J FROM 1 BY 1 UNTIL J = I.
010       SELECT-AIN.
010           IF KAIN EQUAL TO ZERO GO TO SELECT-CTY.
010           MOVE TEMP(KAIN) TO OUT-REC  GO TO LAST-SEL.
010       SELECT-CTY.
010           IF KCTY = ZERO GO TO NEXT-TO-LAST.
010           MOVE TEMP(KCTY) TO OUT-REC  GO TO LAST-SEL.
010       SELECT-AIN-CTY.
010           IF TLACE(J) = 'AIN' COMPUTE KAIN = J.
010           IF TLACE(J) = 'CTY' COMPUTE KCTY = J.
010       NEXT-TO-LAST.
010           MOVE TEMP(1) TO OUT-REC.
010       LAST-SEL.
010           MOVE ZERO TO KCTY, KAIN.
010           WRITE OUT-REC ADD 1 TO REC-COUNT WRITE PRINT-LINE
010               FROM OUT-REC.
010       END-O-SORT.
010            PERFORM SELECT-LOCNA THRU LAST-SEL.
010           MOVE REC-COUNT TO O-REC-COUNT.
010           WRITE PRINT-LINE FROM LAST-LINE.
010       END-RUN SECTION.
010           CLOSE PRINT-FILE OUT-FILE ARLOC-FILE.
010           STOP RUN.
```

END ELT.

@HDG MRG PRE-PROCESSOR PROGRAM MORSAROLL ••UNCLASSIFIED•• •L•0

```
@ELT,L   24PRG.MORSA-ROLL
ELT007 RLIB70 08/16-07:40:54-(17,)
013      IDENTIFICATION DIVISION.                                     00000100
013      PROGRAM-ID.                          MORSA-ROLL.             00000200
013      AUTHOR.                              E MONTAGNE.             00000300
013      DATE-WRITTEN.                        4 NOV  75.              00000400
013      REMARKS.  PROGRAM READS MORSA FILE, ELIMINATES ARMY RECORDS  00000500
013          AND INTRA-THEATER RECORDS, SORTS ON RDD, DESTINATION,    00000600
013          ORIGIN, TRAVEL MODE AND AVAIL DATE.  THE REMAINING       00000700
013          RECORDS ARE PACKAGED TO PRODUCE SMOBSMOD CARDS.          00000800
013      ENVIRONMENT DIVISION.                                        00000900
013      CONFIGURATION  SECTION.                                      00001000
013      SOURCE-COMPUTER.  UNIVAC-1108.                               00001100
013      OBJECT-COMPUTER.  UNIVAC-1108.                               00001200
013      INPUT-OUTPUT SECTION.                                        00001300
013      FILE-CONTROL.                                                00001400
013          SELECT PACKAGE-FILE ASSIGN TO PRINTER 24PACKAGE.
013          SELECT MORSA-FILE ASSIGN TO UNISERVO 24MORSA-TP.         00001500
013          SELECT PRINT-FILE ASSIGN TO PRINTER.                     00001600
013          SELECT O-NODE-FILE  ASSIGN TO CARD-READER.               00001800
013          SELECT D-NODE-FILE  ASSIGN TO CARD-READER.               00001900
013          SELECT SORT-ING     ASSIGN TO MASS-STORAGE XA.           00002000
013           SELECT CARD-FILE ASSIGN TO CARD-READER.
013      DATA DIVISION.                                               00002100
013      FILE SECTION.                                                00002200
013      FD  CARD-FILE  LABEL RECORD OMITTED.
013      01  CARD-IN  PIC X(80).
013      FD  PACKAGE-FILE LABEL RECORD OMITTED.
013      01  PACK-REC  PIC X(132).
013      FD  MORSA-FILE BLOCK 6 RECORDS   RECORDING MODE 1            00002300
013                LABEL RECORD OMITTED.                              00002400
013      01  MORSA-REC.                                               00002500
013          02  CNTL.                                                00002600
013              03  PLAN-ID          PIC X.                          00002700
013              03  SERV             PIC X.                          00002800
013              03  SEQ              PIC 9(4).                       00002900
013              03  TT       PIC X.                                  00003000
013          02  AGG-SEQ      PIC X(4).                               00003100
013          02  RT           PIC  X.                                 00003200
013          02  RD       PIC X.                                      00003300
013          02  REC-NAME     PIC X(12).                              00003400
013          02  LEVEL        PIC X(3).                               00003500
013          02  T-MODE.                                              00003600
013              03  T-MODE1      PIC X.                              00003700
013              03  T-MODE2-3    PIC XX.                             00003800
013          02  AVAIL        PIC 999.                                00003900
013          02  RDD          PIC 999.                                00004000
013          02  PORT-DATE    PIC X(3).                               00004100
013          02  NODE-DATE    PIC X(3).                               00004200
013          02  ORIGIN-SET.                                          00004300
013              03  NAME-O       PIC X(14).                          00004400
013              03  INST-O       PIC X(3).                           00004500
013              03  CODE-2-O     PIC XX.                             00004600
013              03  CODE-5-O     PIC X(5).                           00004700
013              03  GEO-O        PIC X(4).                           00004800
013          02  POD.                                                 00004900
013              03  NAME-P       PIC X(14).                          00005000
```

```
013                 03  INST-P       PIC X(3).                          00005100
013                 03  CODE-2-P     PIC XX.                            00005200
013                 03  CODE-5-P     PIC X(5).                          00005300
013                 03  GEO-P        PIC X(4).                          00005400
013             02  DESTINATION-SET.                                    00005500
013                 03  NAME-D     PIC X(14).                           00005600
013                 03  INST-D       PIC X(3).                          00005700
013                 03  CODE-2-D     PIC XX.                            00005800
013                 03  CODE-5-D     PIC X(5).                          00005900
013                 03  GEO-D        PIC X(4).                          00006000
013         * * * DATA SCALED BY .001 FOR SMOBSMOD.   * * *             00006100
013         * * * MORSA UNITS ARE UNIT PAX, TONS OR SQ FT               00006200
013         * * * SMOBSMOD UNITS ARE THOUSANDS* * *                     00006300
013             02  PAX            PIC 999V999.                         00006400
013             02  A-AMMO         PIC 999V999.                         00006500
013             02  A-SUP          PIC 999V999.                         00006600
013             02  BULK           PIC 999V999.                         00006700
013             02  OVER-POL       PIC 999V999.                         00006800
013             02  OUT-SZ         PIC 999V999.                         00006900
013             02  NAT            PIC 999V999.                         00007000
013             02  TOTAL-WT       PIC 999V999.                         00007100
013             02  TPSN       PIC X(5).                                00007200
013             02  SRC        PIC X(11).                               00007300
013             02  FRN        PIC X(4).                                00007400
013             02  UTC        PIC X(5).                                00007500
013             02  UIC        PIC X(6).                                00007600
013             02  AC-NR      PIC X(3).                                00007700
013             02  FILLER     PIC X(10).                               00007800
013             02  TIME-SAVE.
013                 03  SAVE-RDD    PIC 999.
013                 03  SAVE-AVAIL PIC 999.
013             02  NSDA-SQ-FT     PIC 999V999.
013             02  FILLER     PIC X(12).                               00008000
013             02  PREPO        PIC X(3).                              00008100
013             02  OO-NODE      PIC 999.
013             02  DD-NODE      PIC 999.
013             02  SVMD     PIC X.
013             02  FILLER     PIC X(35).                               00008200
013         FD  PRINT-FILE LABEL RECORD OMITTED.                        00008300
013         01  PRINT-LINE    PIC X(132).                               00008400
013         SD  SORT-ING.                                               00008700
013         01  SORT-REC.                                               00008800
013             02  FILLER          PIC X(28).                          00008900
013             02  T-MODE1         PIC X.                              00009000
013             02  FILLER     PIC XX.                                  00009100
013             02  AVAIL           PIC 999.                            00009200
013             02  RDD             PIC 999.                            00009300
013             02  FILLER          PIC X(23).                          00009400
013             02  CODE-2-O        PIC XX.                             00009500
013             02  FILLER          PIC X(54).                          00009600
013             02  CODE-2-D        PIC XX.                             00009700
013             02  FILLER     PIC X(128).
013             02  OO-NODE      PIC 999.
013             02  DD-NODE      PIC 999.
013             02  FILLER          PIC X(36).
013         FD  O-NODE-FILE  LABEL RECORD OMITTED.                      00009900
013         01  O-NODE-REC.                                             00010000
```

```
013          02  ORIG     PIC XX.                                        00010100
013          02  FILLER   PIC X.                                         00010200
013          02  ORIG-NODE PIC 999.                                      00010300
013          02  FILLER   PIC X(74).                                     00010400
013      FD  D-NODE-FILE  LABEL RECORD OMITTED.                          00010500
013      01  D-NODE-REC.                                                 00010600
013          02  DEST     PIC XX.                                        00010700
013          02  FILLER   PIC X.                                         00010800
013          02  DEST-NODE PIC 9(3).                                     00010900
013          02  FILLER   PIC X(74).                                     00011000
013      COMMON-STORAGE SECTION.
013      77  CARDS-ABC      PIC X(84).
013      01  ARRAY-1.
013          03  PKG            PIC 9(6).
013          03  RDD            PIC 999.
013          03  AVAIL          PIC 999.
013          03  DD-NODE        PIC 999.
013          03  OO-NODE        PIC 999.
013          03  MODE-TS        PIC X.
013          03  PAX            PIC 9(5)V999.
013          03  NAT            PIC 9(5)V999.
013          03  NSDA-NAT       PIC 9(5)V999 VALUE ZERO.
013          03  NSDA-SQ-FT     PIC 9(5)V999.
013          03  VEH-OUT                  PIC 9(5)V999     VALUE ZERO.
013          03  POL                      PIC 9(5)V999.
013          03  FILLER                   PIC X(29)  VALUE SPACES.
013          03  OUT-SZ                   PIC 9(5)V999.
013          03  NSDA-OVR-C               PIC 9(5)V999    VALUE ZERO.
013          03  NSDA-OVR-NC              PIC 9(5)V999    VALUE ZERO.
013          03  VEH-OVR-C              PIC 9(5)V999    VALUE ZERO.
013          03  VEH-OVR-NC             PIC 9(5)V999    VALUE ZERO.
013          03  OVR                    PIC 9(5)V999.
013          03  NV-OVR-C             PIC 9(5)V999    VALUE ZERO.
013          03  VEH-BULK-C            PIC 9(5)V999    VALUE ZERO.
013          03  BULK1                 PIC 9(5)V999.
013          03  DRY-RESUP             PIC 9(5)V999    VALUE ZERO.
013          03  RESUP-AMMO            PIC 9(5)V999.
013          03  FILLER                PIC X(8)      VALUE '999999  '.
013      WORKING-STORAGE SECTION.                                        00011100
013      01  C-CARD.
013          02  FILLER      PIC X(8)    VALUE '   5444.'.
013          02  FILLER      PIC X(22)   VALUE SPACES.
013          02  FILLER      PIC XX      VALUE '0.'.
013          02  FILLER      PIC X(8)    VALUE SPACES.
013          02  FILLER      PIC XX      VALUE '0.'.
013          02  FILLER      PIC X(38)    VALUE SPACES.
013      01  HEADER-1.
013          02  FILLER   PIC X(30) VALUE 'ORIGIN             LEV NAME     '.
013          02  FILLER   PIC X(30) VALUE '     RDD DESTINATION       M RTR'.
013          02  FILLER   PIC X(30) VALUE 'D AVL PAX     A-AMMO A-SUP  BUL'.
013          02  FILLER   PIC X(30) VALUE 'K   OV/POL OUT     NAT    TOTAL'.
013          02  FILLER   PIC X(12) VALUE '  NSDA-SQFT '.
013      01  LINE-1.
013          02  ORIGIN-SET.
013          03  NAME-O  PIC X(14).
013              03  CODE-2-O   PIC XX.
013          02  FILLER    PIC X  VALUE SPACE.
```

```
013         02  LEVEL     PIC X(3).
013         02  FILLER    PIC X  VALUE SPACE.
013         02  REC-NAME  PIC X(12).
013         02  FILLER    PIC X  VALUE SPACES.
013         02  RDD       PIC 999.
013         02  FILLER PIC X VALUE SPACES.
013         02  DESTINATION-SET.
                03  NAME-D     PIC X(14).
                03  CODE-2-D   PIC XX.
013         02  FILLER    PIC X  VALUE SPACE.
013         02  T-MODE.
                03  T-MODE1    PIC X.
013         02  FILLER PIC X  VALUE SPACES.
013         02  RT        PIC X.
013         02  RD        PIC X.
013         02  FILLER PIC     XXX  VALUE SPACE.
013         02  AVAIL PIC 999.
013         02  FILLER    PIC X  VALUE SPACE.
013         02  PAX    PIC 999V999.
013         02  FILLER PIC X  VALUE SPACES.
013         02  A-AMMO PIC 999V999.
013         02  FILLER PIC X  VALUE SPACES.
013         02  A-SUP  PIC 999V999.
013         02  FILLER PIC X  VALUE SPACES.
013         02  BULK   PIC 999V999.
013         02  FILLER PIC X  VALUE SPACES.
013         02  OVER-POL PIC 999V999.
013         02  FILLER PIC X  VALUE SPACES.
013         02  OUT-SZ   PIC 999V999.
013         02  FILLER PIC X  VALUE SPACES.
013         02  NAT    PIC 999V999.
013         02  FILLER PIC X  VALUE SPACES.
013         02  TOTAL-WT  PIC 999V999.
013         02  FILLER PIC X  VALUE SPACES.
013         02  NSDA-SQ-FT  PIC 999V999.
013         02  FILLER  PIC X(4)  VALUE SPACES.
013     01  A-CARD.                                                     00011200
013         02  FILLER       PIC X VALUE SPACE.                         00011300
013         02  FILLER       PIC X(20)  VALUE 'OTHER SERVICE PKG NR'.   00011400
013         02  PKG-NR-O     PIC ZZZZ9.                                 00011500
013         02  FILLER       PIC X(17)  VALUE '          RDD=  0'.      00011600
013         02  RDD            PIC 9(3).                                00011700
013         02  FILLER       PIC X(15)  VALUE '00       AVL=  0'.       00011800
013         02  AVAIL          PIC 9(3).                                00011900
013         02  FILLER PIC XX VALUE '00'.                               00011950
013         02  NEG          PIC X(14)  VALUE SPACES.                   000120
013     01  O-NODE-TABLE.                                               00012100
013         02  TABLE-O  OCCURS 20 TIMES.                               00012200
                03  ORIGIN     PIC XX.                                  00012300
013             03  O-NODE     PIC 999.
013     01  D-NODE-TABLE.                                               00012500
013         02  TABLE-D  OCCURS 20 TIMES.                               00012600
013             03  DESTIN     PIC XX.                                  00012700
013             03  D-NODE     PIC 999.
013     01  CARGO-TYPE-TABLE.                                           00012900
013         02  FILLER       PIC 99  VALUE  1.                          00013000
013         02  FILLER       PIC 99  VALUE 15.
```

```
013          02  FILLER       PIC 99  VALUE 12.                          00013200
013          02  FILLER       PIC 99  VALUE  6.                          00013300
013          02  FILLER       PIC 99  VALUE  7.                          00013400
013          02  FILLER       PIC 99  VALUE  2.                          00013500
013          02  FILLER       PIC 99  VALUE 15.                          00013600
013          02  FILLER       PIC 99  VALUE 15.                          00013700
013          02  FILLER       PIC 99  VALUE  4.
013          02  FILLER       PIC 99   VALUE 17.
013      01  TABLE-CARGO-TYPE REDEFINES CARGO-TYPE-TABLE.                00013900
013          02  CARGO-TYPE  PIC 99 OCCURS 10 TIMES.
013      01  NO-NODE-LINE.                                               00014100
013          02  FILLER   PIC X(19)  VALUE '•••NODE NOT FOUND   '.       00014200
013          02  CODE-2  PIC XX.                                         00014300
016          02  FILLER  PIC X        VALUE SPACES.
013          02  FLAG-1   PIC 9 VALUE 2.
016          02  FILLER   PIC X        VALUE SPACES.
016          02  TOTAL-WT       PIC 999V999.
016          02  BULK           PIC 999V999.
016          02  PAX            PIC 999V999.
016          02  RDD            PIC 999.
016          02  AVAIL          PIC 999.
016          02  FILLER         PIC X(84).
013      01  COM-PARE.                                                   00014500
013          02  T-MODE.                                                  00014700
013              03  T-MODE1    PIC X.                                   00014800
013          02  OO-NODE  PIC 999.
013          02  DD-NODE  PIC 999.
013          02  RDD      PIC 999.
013          02  AVAIL    PIC 999.
013      01  SAVE-DATA.                                                  00015400
013          02  COMPARE-SET.                                            00015500
013          03  T-MODE.
013                  04  T-MODE1  PIC X.                                    00015700
013             03  OO-NODE  PIC 999.
013             03  DD-NODE  PIC 999.
013              03  RDD       PIC 999.
013              03  AVAIL     PIC 999.
013          02  ADD-SET  USAGE IS COMPUTATIONAL.                        00016100
013              03  PAX        PIC 9(5)V999 VALUE ZERO.                 00016200
013              03  BULK1        PIC 9(5)V999 VALUE ZERO.                00016300
013              03  OVR    PIC 9(5)V999 VALUE ZERO.                     00016400
013              03  POL        PIC 9(5)V999 VALUE ZERO.                 00016500
013              03  OUT-SZ   PIC 9(5)V999 VALUE ZERO.                   00016600
013              03  NAT        PIC 9(5)V999 VALUE ZERO.                 00016700
013              03  A-AMMO     PIC 9(5)V999 VALUE ZERO.                 00016800
013              03  A-SUP      PIC 9(5)V999 VALUE ZERO.                 00016900
013              03  NSDA-SQ-FT       PIC 9(5)V999 VALUE ZERO.           00017000
013              03  RESUP-AMMO     PIC  9(5)V999    VALUE ZERO.
013          02  QTY-TABLE  REDEFINES ADD-SET USAGE IS COMP.             00017100
013              03  QTY          PIC 9(5)V999  OCCURS 10 TIMES.
013      01  DELTA-CARD.
013          02  DELTA-RDD        PIC 99.
013          02  DELTA-AVAIL      PIC 99.
013          02  FILLER           PIC X(76).
013      01  B-CARD.                                                     00017300
013          02  FILLER       PIC X(4)  VALUE SPACES.                    00017400
013          02  AVAIL        PIC 999.                                   00017500
```

```
013           02  FILLER      PIC X(5)   VALUE '•     '.               00017600
013           02  DEST-NODE   PIC 999.                                 00017700
013           02  FILLER      PIC XX     VALUE SPACES.                 00017800
013           02  ORIG-NODE   PIC 999.                                 00017900
013           02  FILLER      PIC X(4)   VALUE SPACES.                 00018000
013           02  T-MODE      PIC  X  VALUE ZERO.                      00018100
013           02  FILLER      PIC X      VALUE SPACES.                 00018200
013           02  PAX         PIC  9(5).999 VALUE ZERO.                00018300
013           02  FILLER      PIC X      VALUE SPACES.                 00018400
013           02  AREG        PIC  9(5).999  VALUE ZERO.               00018500
013           02  CARGO-TYPE-FBLK  PIC  ZZZZ9.                         00018600
013           02  FILLER      PIC X(25)  VALUE SPACES.                 00018700
013       77  J  PICTURE 99 VALUE 1  USAGE IS COMP.
013       77  PKG-NR  PIC 9(5) USAGE IS COMPUTATIONAL VALUE ZERO.      00018900
013       77  COMP-POL PIC 9(6)V999 USAGE IS COMP.
013       77  O-CNT     PIC 9(4) USAGE IS COMPUTATIONAL VALUE ZERO.    00019100
013       77  D-CNT     PIC 9(4) USAGE IS COMPUTATIONAL VALUE ZERO.    00019200
013       77  LINE-COUNT  PIC 99 USAGE IS COMP VALUE 54.
013       77  HALF-DELTA-AVAIL        PIC 99    USAGE IS COMP.
013       77  HALF-DELTA-RDD      PIC 99    USAGE IS COMP.
017       77  POL-WT          PIC 999V999 USAGE IS COMP.
013       PROCEDURE DIVISION.                                          00019300
013           OPEN INPUT MORSA-FILE O-NODE-FILE.                       00019400
013           OPEN OUTPUT  PRINT-FILE PACKAGE-FILE.
013      •••• READ NODE FILES AND SET UP NODE TABLES•••                00019600
013       READ-O-NODE.                                                 00019700
013           READ O-NODE-FILE AT END GO TO READ-D-NODE.               00019800
013           ADD 1 TO O-CNT MOVE ORIG-NODE IN O-NODE-REC              00019900
013            TO O-NODE(O-CNT)  MOVE ORIG  TO ORIGIN(O-CNT).
013           GO TO READ-O-NODE.                                       00020100
013       READ-D-NODE.                                                 00020200
013           CLOSE O-NODE-FILE OPEN INPUT D-NODE-FILE.                00020300
013           ADD 1 TO O-CNT.                                          00020400
013       READ-DD-NODE.                                                00020500
013           READ D-NODE-FILE AT END GO TO END-D.                     00020600
013           ADD 1 TO D-CNT MOVE DEST-NODE IN D-NODE-REC              00020700
013                   TO D-NODE(D-CNT).
013            MOVE DEST TO DESTIN(D-CNT).                             00020900
013            GO TO READ-DD-NODE.                                     00021000
013       END-D.                                                       00021100
013           ADD 1 TO D-CNT.
013           CLOSE D-NODE-FILE.
013           OPEN INPUT CARD-FILE.
013           READ CARD-FILE INTO DELTA-CARD AT END NEXT SENTENCE.
013           COMPUTE HALF-DELTA-AVAIL = DELTA-AVAIL / 2.
013           COMPUTE HALF-DELTA-RDD   = DELTA-RDD   / 2.
013       A-SORT SECTION.
013            SORT SORT-ING ON ASCENDING KEY RDD AVAIL T-MODE1
013                      OO-NODE DD-NODE
013           INPUT PROCEDURE IS SORT-1                                00021600
013           OUTPUT PROCEDURE IS SMOBSMOD.                            00021700
013           GO TO END-RUN.                                           00021800
013       SORT-1 SECTION.                                              00021900
013       READ-MORSA.                                                  00022000
013      •                                                             00022100
013      •  ELIMINATE ARMY RECORDS AND INTRA THEATER RECORDS••••       00022200
013      •                                                             00022300
```

```
013        READ MORSA-FILE AT END GO TO END-SORT-1.                  00022400
013        IF SERV IN MORSA-REC = 'A' GO TO READ-MORSA.              00022500
013    ***** DEBUG  **************
013    *     IF RDD IN MORSA-REC GREATER THAN 45 GO TO READ-MORSA.
013    ***** DEBUG  **************
013        IF T-MODE1 IN MORSA-REC = TO 'A' OR 'P' OR 'S'
013                    PERFORM SET-NODES.
013                ELSE GO TO READ-MORSA.
013        IF FLAG-1 NOT LESS THAN 2 PERFORM ROLL-ON-DELTA.
013        GO TO READ-MORSA.                                          00022900
013    ROLL-ON-DELTA.
013    * * * RDD * * *
013        MOVE RDD  IN MORSA-REC TO SAVE-RDD
013        MOVE AVAIL IN MORSA-REC TO SAVE-AVAIL.
013        DIVIDE DELTA-RDD INTO RDD IN MORSA-REC.
013        DIVIDE DELTA-AVAIL INTO AVAIL IN MORSA-REC.
013    * * *MODE * * *
013        MOVE T-MODE1 IN MORSA-REC TO SVMD.
013        IF SVMD = 'P' MOVE 'U' TO T-MODE1 IN MORSA-REC.
013        IF SVMD = 'S'
013                  IF PAX IN MORSA-REC > 0 MOVE '3' TO T-MODE1
013                            IN MORSA-REC
013                   ELSE MOVE '2' TO T-MODE1 IN MORSA-REC.
013        IF SVMD = 'A'
013                  IF NAT IN MORSA-REC = 0 MOVE '1' TO T-MODE1
013                            IN MORSA-REC
013                   ELSE IF PAX IN MORSA-REC = 0 MOVE '2' TO
013                             T-MODE1 IN MORSA-REC
013                           ELSE MOVE '3' TO T-MODE1 IN MORSA-REC.
013        IF RT IN MORSA-REC = 'P' AND OVER-POL IN MORSA-REC > 0
013                PERFORM CHECK-FOR-PKG-POL
013                    ELSE RELEASE SORT-REC FROM MORSA-REC.
013    END-SORT-1.                                                    00023000
013         CLOSE MORSA-FILE.                                         00023100
013    * * *NODES*  * *                                               00026300
013    SET-NODES SECTION.
013        MOVE 2 TO FLAG-1.
013    O-NODES.                                                       00026400
013        IF J = O-CNT MOVE CODE-2-O IN MORSA-REC TO CODE-2          00026500
017              MOVE CORR MORSA-REC TO NO-NODE-LINE
013           MOVE 1 TO FLAG-1  WRITE PRINT-LINE FROM NO-NODE-LINE
013                                GO TO J-1.
013        IF CODE-2-O IN MORSA-REC = ORIGIN(J) MOVE                  00026800
013                O-NODE(J) TO OO-NODE IN MORSA-REC GO TO J-1
013           ELSE ADD 1 TO J GO TO O-NODES.                          00027000
013    J-1.                                                           00027100
013        MOVE 1 TO J.                                               00027200
013    D-NODES.                                                       00027300
013        IF J = D-CNT MOVE CODE-2-D IN MORSA-REC TO CODE-2          00027400
016          MOVE CORR MORSA-REC TO NO-NODE-LINE
013          MOVE 0 TO FLAG-1  WRITE PRINT-LINE FROM NO-NODE-LINE
013                                GO TO SKIP-REC.
013        IF CODE-2-D IN MORSA-REC = DESTIN(J) MOVE D-NODE(J)
013                          TO DD-NODE IN MORSA-REC GO TO SKIP-REC
013           ELSE ADD 1 TO J GO TO D-NODES.                          00027900
013    ***  DO NOT PROCESS RECORD IF NODE NOT FOUND***
013    SKIP-REC.
```

```
013            MOVE 1 TO J.
013        SMOBSMOD SECTION.                                                    00023200
013              RETURN SORT-ING RECORD INTO MORSA-REC AT END GO                00023300
013                      TO END-S. PERFORM RE-SET-DATES.                        000234
013              PERFORM SAVE-INFO.
013        RETURN-AGAIN.                                                        00023600
013             RETURN SORT-ING RECORD INTO MORSA-REC AT END GO                 00023700
013                      TO END-S.                                              00023800
013        RE-SET-DATES.
013             COMPUTE AVAIL IN MORSA-REC  =  AVAIL IN MORSA-REC •
013                        DELTA-AVAIL + HALF-DELTA-AVAIL.
013             COMPUTE RDD   IN MORSA-REC  =  RDD   IN MORSA-REC •
013                        DELTA-RDD    + HALF-DELTA-RDD.
013        P-1.
013             MOVE CORRESPONDING MORSA-REC TO COM-PARE.                       00023900
013             IF COM-PARE = COMPARE-SET PERFORM ADD-PROC                      00024U
013                     ELSE PERFORM SMOBSMOD-PROC THRU END-SMOB.
013             GO TO RETURN-AGAIN.
013        ADD-PROC.                                                            00024300
013              PERFORM PACKAGE-DETAIL.
013             ADD CORR MORSA-REC TO ADD-SET.                                  00024400
013             IF RT IN MORSA-REC = 'P' MOVE OVER-POL IN MORSA-REC TO
013                     COMP-POL,
013                        COMPUTE COMP-POL = COMP-POL • 1000,                  00024600
013                        ADD COMP-POL TO POL IN SAVE-DATA                     00024700
013                ELSE ADD OVER-POL IN MORSA-REC TO OVR IN SAVE-DATA.
013             IF RT IN MORSA-REC = 'S' AND RD IN MORSA-REC = 'A'
013                    ADD BULK IN MORSA-REC TO RESUP-AMMO IN SAVE-DATA,
013                     ELSE ADD BULK IN MORSA-REC TO BULK1 IN SAVE-DATA.
013        SMOBSMOD-PROC.                                                       00024900
013       •   •   •   A CARD  • • • •                                           00025000
013             ADD 1 TO PKG-NR MOVE PKG-NR TO PKG-NR-0.                        00025100
013             MOVE CORR COMPARE-SET TO A-CARD.
013              WRITE PRINT-LINE FROM A-CARD AFTER 2.
013             MOVE A-CARD TO CARDS-ABC.
013             ENTER FORTRAN SMOBS SUBROUTINE.
013       • • •  BOTTOM LINE PACKAGE DEATAIL • • •
013             WRITE PACK-REC FROM A-CARD MOVE SPACES TO PACK-REC
013             WRITE PACK-REC  ADD 2 TO LINE-COUNT.
013       • • • • B CARDS • • • •                                               00025500
013       • • •AVAIL• • •                                                       00025600
013             MOVE AVAIL IN A-CARD     TO AVAIL IN B-CARD.                    000257
013             MOVE T-MODE1 IN SAVE-DATA TO T-MODE IN B-CARD.
013             MOVE DD-NODE IN SAVE-DATA TO DEST-NODE IN B-CARD.
013             MOVE OO-NODE IN SAVE-DATA TO ORIG-NODE IN B-CARD.
013       ••• PAX •••                                                           00028100
013             IF PAX IN SAVE-DATA NOT GREATER THAN ZERO GO TO CARGO.          00028200
013              MOVE PAX IN SAVE-DATA TO PAX IN B-CARD. MOVE 1 TO              00028300
013              CARGO-TYPE-FBLK.  WRITE PRINT-LINE FROM B-CARD.
013              MOVE B-CARD TO CARDS-ABC.
013             ENTER FORTRAN SMOBS SUBROUTINE.
013             MOVE ZERO TO PAX IN B-CARD.                                     00028550
013       •••CARGO•••                                                           00028600
013        CARGO.                                                               00028700
013              PERFORM A-REG VARYING J FROM 2 BY 1 UNTIL J = 11.              00028800
013             MOVE ZERO TO AREG  IN B-CARD.                                   00028850
013       • • •  C-CARD • • •
```

```
013         MOVE C-CARD TO CARDS-ABC.
013         ENTER FORTRAN SMOBS SUBROUTINE.
013     • • • WRITE ARRAY FOR PATTY• • •
013         MOVE CORR ADD-SET TO ARRAY-1  MOVE CORR COMPARE-SET
013                 TO ARRAY-1.  MOVE PKG-NR TO PKG.
013                 MOVE T-MODE IN B-CARD TO MODE-TS.
013                 ENTER FORTRAN SMOBS2 SUBROUTINE.
013         MOVE 1 TO J.
013     SAVE-INFO.                                                      00029000
013         PERFORM PACKAGE-DETAIL.
013         MOVE CORR MORSA-REC TO COMPARE-SET.
013         MOVE CORR MORSA-REC TO ADD-SET.
013         IF RT IN MORSA-REC = 'S' AND RD IN MORSA-REC = 'A'
013          MOVE BULK IN MORSA-REC TO RESUP-AMMO IN SAVE-DATA
013              MOVE ZERO TO BULK1 IN SAVE-DATA
013               ELSE MOVE BULK IN MORSA-REC TO BULK1 IN SAVE-DATA
013                   MOVE  ZERO TO RESUP-AMMO IN SAVE-DATA.
013         IF RT IN MORSA-REC = 'P' MOVE OVER-POL IN                   00029300
013                 MORSA-REC TO COMP-POL COMPUTE COMP-POL =            00029320
013                 COMP-POL • 1000  MOVE COMP-POL TO POL IN SAVE-DATA  00029350
013                  MOVE ZERO TO OVR IN SAVE-DATA
013             ELSE MOVE OVER-POL IN MORSA-REC TO OVR IN SAVE-DATA     00029500
013             MOVE ZERO TO POL IN SAVE-DATA.
013     END-SMOB.
013         EXIT.
013     A-REG.                                                          00029700
013          IF QTY(J) GREATER THAN ZERO, MOVE QTY(J) TO                00029800
013              AREG IN B-CARD, MOVE CARGO-TYPE(J) TO                  00029900
013              CARGO-TYPE-FBLK, MOVE B-CARD TO CARDS-ABC,
013              ENTER FORTRAN SMOBS SUBROUTINE.
013              WRITE PRINT-LINE FROM B-CARD.                          00030100
013     END-S.                                                          00030200
013          PERFORM SMOBSMOD-PROC THRU CARGO.                          00030300
013         MOVE C-CARD TO CARDS-ABC.
013         ENTER FORTRAN SMOBS1 SUBROUTINE.
013     END-RUN SECTION.                                                00030400
013          CLOSE PRINT-FILE  PACKAGE-FILE.
013         CLOSE CARD-FILE.
013             STOP RUN.                                               00030600
013     ••• EACH RECORD MAKING UP A PACKAGE IS PRINTED•••
013     PACKAGE-DETAIL SECTION.
013     CHECK-COUNT.
013         IF LINE-COUNT LESS THAN 54 GO TO PRINT-DETAIL.
013     SET-HEADER.
013         MOVE ZERO TO LINE-COUNT WRITE PACK-REC FROM
013                 HEADER-1 AFTER ADVANCING TOP-OF-PAGE.
013         MOVE SPACES TO PACK-REC  WRITE PACK-REC.
013     PRINT-DETAIL.
013         MOVE CORR MORSA-REC TO LINE-1.
013         MOVE SVMD TO T-MODE1 IN LINE-1.
013         MOVE SAVE-RDD TO RDD IN LINE-1.
013         MOVE SAVE-AVAIL TO AVAIL IN LINE-1.
013          WRITE PACK-REC FROM LINE-1.
013                     ADD 1 TO LINE-COUNT.
013     SEPARATE-POL SECTION.
013     CHECK-FOR-PKG-POL.
013           IF BULK IN MORSA-REC > 0 PERFORM MOVE-2 THRU MOVE-1
```

```
013                      ELSE PERFORM MOVE-1.
013        MOVE-2.
013       • • • MAKE SEPARATE RECORD FOR PACKAGE POL• • •
013            MOVE OVER-POL IN MORSA-REC TO COMP-POL.
013            MOVE ZERO TO OVER-POL IN MORSA-REC.
013       • • •1000/7.213 BBLS PER ST = 138.639 ST PER 1000 BBLS• • •
013            COMPUTE POL-WT = 138.639 • COMP-POL.
014            SUBTRACT POL-WT FROM TOTAL-WT IN MORSA-REC.
013            RELEASE SORT-REC FROM MORSA-REC.
013            MOVE POL-WT TO TOTAL-WT IN MORSA-REC MOVE COMP-POL TO
013                      OVER-POL IN MORSA-REC.
015            MOVE ZERO TO BULK IN MORSA-REC.
013       • • • MAKE RECORD FOR POL.  SET NODE = 191 FOR GALVESTON.• • •
013        MOVE-1.
013            MOVE 191 TO OO-NODE IN MORSA-REC.
013            RELEASE SORT-REC FROM MORSA-REC.

END ELT.


@ELT.L   24PRG.SMOBS
ELT007 RLIB70 08/16-07:40:57-(2,)
000           SUBROUTINE SMOBS
002           COMMON DUMMY, CARDS(14),A1(16),A2(16)
000     102   FORMAT (14A6)
000           WRITE (2,102) CARDS
000           RETURN
002           ENTRY SMOBS2
002           WRITE (3,103) A1
002           WRITE (3,103) A2
002     103   FORMAT (16A6)
002           RETURN
000           ENTRY SMOBS1
000           WRITE (2,102) CARDS
000           WRITE (2,102) CARDS
000           ENDFILE 2
002           ENDFILE 3
000           RETURN
000           END

END ELT.

@HDG    MRG  PRE-PROCESSOR PROGRAM TUCHAROLL     ••UNCLASSIFIED•• .L,0
```

```
@ELT,L    32MARTIN.TUSUM
ELT007 RL1B70 08/16-07:40:58-(33,)
       029            IDENTIFICATION DIVISION.
       029            PROGRAM-ID.       LOAD-FIF2.
       029            AUTHOR.           G.L.MARTIN
       029            DATE-WRITTEN.     JANUARY 1975
       029            DATE-COMPILED.
       029            INSTALLATION.     US ARMY CAA.
       029            SECURITY.         UNCLASSIFIED.
       029            REMARKS.          PROGRAM LOADS CAMP FIF2 FILE FROM A
       029                NONSTANDARD TUCHA FILE WITH SRC APPENDED TO THE
       029                ABF1 RECORD.
       029            ENVIRONMENT DIVISION.
       029            CONFIGURATION SECTION.
       029            SOURCE-COMPUTER. UNIVAC-1108.
       029            OBJECT-COMPUTER. UNIVAC-1108.
       029            INPUT-OUTPUT SECTION.
       029            FILE-CONTROL.
       029                SELECT PRINT-FILE ASSIGN TO PRINTER.
       029                SELECT CARD-FILE ASSIGN TO CARD-READER.
       029                SELECT TUCHA ASSIGN TO UNISERVO TUCHA.
       029                SELECT SAVE-FILE ASSIGN TO UNISERVO SAVE.
       029                SELECT SORT-FILE ASSIGN TO MASS-STORAGE XA.
       029                SELECT F5 ASSIGN TO MASS-STORAGE 32FIF2
       029                        ORGANIZATION IS INDEXED
       029                        FILE-DESCRIPTION IS F5-DESC
       029                        ACCESS MODE IS RANDOM
       029                        SYMBOLIC KEY IS F5KEY.
       029            DATA DIVISION.
       029            FILE SECTION.
       029            FD  PRINT-FILE
       029                LABEL RECORDS ARE OMITTED
       029                DATA RECORD IS PRINT-LINE.
       029            01  PRINT-LINE            PICTURE IS X(132).
       029            FD  CARD-FILE
       029                LABEL RECORDS ARE OMITTED
       029                DATA RECORD IS INPUT-CARD.
       029            01  INPUT-CARD            PICTURE IS X(80).
       029            FD  SAVE-FILE
       029                LABEL RECORDS ARE OMITTED
       029                BLOCK CONTAINS 24 RECORDS
       029                RECORD CONTAINS 138 CHARACTERS
       029                DATA RECORD IS SAVE-REC.
       029            01  SAVE-REC              PICTURE IS X(138).
       029            FD  TUCHA
       029                LABEL RECORDS ARE OMITTED
       031                BLOCK CONTAINS 24 RECORDS
       029                DATA RECORDS ARE F1-REC, F2-REC, F3-REC.
       029            01  F1-REC.
       029                02 ABF1.
       029                03 TYPEA.
       029                05 UTC                PICTURE IS X(05).
       029                05 CARGO.
       029                07 CARGO1             PICTURE IS A(01).
       029                88 BULKPOL            PICTURE IS A(01)   VALUE IS 'G'.
       029                07 CARGO2             PICTURE IS 9(01).
       029                05 CONTAINER          PICTURE IS A(01).
```

```
029          05 TUTYPE          PICTURE IS 9(01).
029          88 F1TYPE          PICTURE IS 9(01)   VALUE IS 1.
029          88 F2TYPE          PICTURE IS 9(01)   VALUE IS 2.
029          88 F3TYPE          PICTURE IS 9(01)   VALUE IS 3.
029          05 F3IDNO          PICTURE IS 9(03).
029          05 ULC             PICTURE IS X(03).
029          05 DEPLOYMENT      PICTURE IS X(01).
029          05 SERVICE         PICTURE IS X(01).
029          05 SECURE          PICTURE IS X(01).
029          05 SHORT-NAME      PICTURE IS X(15).
029          05 VALIDITY        PICTURE IS X(01).
029          88 TYPEA-RCRD      PICTURE IS X(01)   VALUE IS 'T'.
029          05 TYPE-STATUS     PICTURE IS X(01).
029          88 CANCELLED       PICTURE IS X(01)   VALUE IS 'C'.
029          05 F1CLASS         PICTURE IS X(01).
029          03 F1DATA.
029          05 TYPE-NAME       PICTURE IS X(54).
029          05 UIC             PICTURE IS X(06).
029          05 CREATE-DATE     PICTURE IS 9(06).
029          05 CHANGE-DATE     PICTURE IS 9(06).
029          05 AUTH-WAR-STR    PICTURE IS 9(05).
029          05 NOPAX           PICTURE IS 9(05).
029          05 F2REQ           PICTURE IS 9(03).
029          05 NOF2            PICTURE IS 9(03).
029          05 NEWF1           PICTURE IS X(01).
029          02 F1-SRC.
029          05 SRC             PICTURE IS X(12).
033          05 FILLER          PICTURE IS X(01).
029       01 F2-REC.
029          03 TUF2.
029          05 F2RKEY          PICTURE IS X(12).
029          05 F2CLASS         PICTURE IS X(01).
029          05 NEWF2           PICTURE IS X(01).
029          05 HLC             PICTURE IS X(01).
029          05 FILLER          PICTURE IS X(03).
029          05 F2SQFT          PICTURE IS 9(06).
029          05 F2STON          PICTURE IS 9(06).
029          05 F2MTON          PICTURE IS 9(06).
029          05 HBPOL           PICTURE IS 9(06).
029          05 F3REQ           PICTURE IS 9(03).
029          05 NOF3            PICTURE IS 9(03).
029          03 F2-FIL.
029          05 FILLER          PICTURE IS X(90).
029       01 F3-REC.
029          03 TUF3.
029          05 F3RKEY          PICTURE IS X(12).
029          05 F3CLASS         PICTURE IS X(01).
029          05 NEWF3           PICTURE IS X(01).
029          05 F3DESC          PICTURE IS X(14).
029          05 LENGTH          PICTURE IS 9(04).
029          05 WIDTH           PICTURE IS 9(03).
029          05 HEIGHT          PICTURE IS 9(03).
029          05 F3SQFT          PICTURE IS 9(04).
029          05 PIECES          PICTURE IS 9(03).
029          05 F3STON          PICTURE IS 9(06).
029          05 F3MTON          PICTURE IS 9(06).
029          05 FILLER          PICTURE IS X(03).
```

```
029            03 F3-FIL.
029            05 FILLER          PICTURE IS X(78).
029        SD  SORT-FILE          DATA RECORD IS LOG-REC.
029        01  LOG-REC.
029            05 SEQNO           PICTURE IS 9(05).
029            05 CDTYPE.
029            07 CDTYPE1         PICTURE IS A(01).
029            07 CDTYPE2         PICTURE IS 9(01).
029            05 PERTO           PICTURE IS 9(05).
029            05 CRCAT.
029            07 CRCAT1          PICTURE IS X(01).
029            07 CRCAT2          PICTURE IS X(01).
029            05 CONCO           PICTURE IS X(01).
029            05 SQFT            PICTURE IS 9(06).
029            05 STOF            PICTURE IS 9(02).
029            05 STON            PICTURE IS 9(06).
029            05 MTON.
029            07 MTOF            PICTURE IS 9(01).
029            07 WHMTO           PICTURE IS 9(06).
029            07 TEMET           PICTURE IS 9(01).
029            05 HELCO           PICTURE IS A(01).
029            05 SRC.
029            07 SRC19           PICTURE IS X(09).
029            07 SRC10           PICTURE IS X(01).
029            07 SRC11-12        PICTURE IS X(02).
029            05 UTC             PICTURE IS X(05).
029            05 UIC             PICTURE IS X(06).
029            05 FILLER          PICTURE IS X(05).
029            66 LEVCO  RENAMES CRCAT THRU CONCO.
029            66 UNSTR  RENAMES SQFT.
029            66 UNAME  RENAMES STOF  THRU TEMET.
029            66 DPIND  RENAMES HELCO.
029        FD  F5
029            LABEL RECORDS ARE STANDARD
029            DATA RECORD IS RF5.
029        01  RF5                PICTURE IS X(5376).
029        COMMON-STORAGE SECTION.
029        01  LINE1.
029            05 SRC1.
029            07 SRC1-9                PIC X(9).
029            07 SRCPAR                PIC X(2).
029            05 LINE-1                PIC X(1)    VALUE '1'.
029            05 FILLER                PIC X(2)    VALUE SPACES.
029            05 ULC                   PIC X(3).
029            05 FILLER                PIC X(2)    VALUE SPACES.
029            05 UTC                   PIC X(5).
029            05 STR                   PIC Z(5)9.
029            05 STON-A0               PIC Z(5)9.
029            05 STON-B0               PIC Z(5)9.
029            05 STON-B1               PIC Z(5)9.
029            05 STON-A1               PIC Z(5)9.
029            05 STON-J1               PIC Z(5)9.
029            05 STON-B2C              PIC Z(5)9.
029            05 STON-B2D              PIC Z(5)9.
029            05 STON-A2C              PIC Z(5)9.
029            05 STON-A2D              PIC Z(5)9.
029            05 STON-J2D              PIC Z(5)9.
```

```
029                05 STON-J2C             PIC Z(5)9.
029                05 STON-A3C             PIC Z(5)9.
029                05 STON-J3C             PIC Z(5)9.
029                05 STON-J7C             PIC Z(5)9.
029                05 STON-M7C             PIC Z(5)9.
029                05 STON-UE              PIC Z(5)9.
029                05 STON-AS              PIC Z(5)9.
029            01  LINE2.
029                05 SRC2                 PIC X(11).
029                05 LINE-2               PIC X(1)   VALUE '2'.
029                05 NAME                 PIC X(12).
029                05 PAX                  PIC Z(5)9.
029                05 SQFT-A0              PIC Z(5)9.
029                05 SQFT-B0              PIC Z(5)9.
029                05 SQFT-B1              PIC Z(5)9.
029                05 SQFT-A1              PIC Z(5)9.
029                05 FILLER               PIC X(6)   VALUE SPACES.
029                05 SQFT-B2C             PIC Z(5)9.
029                05 SQFT-B2D             PIC Z(5)9.
029                05 SQFT-A2C             PIC Z(5)9.
029                05 SQFT-A2D             PIC Z(5)9.
029                05 FILLER               PIC X(12)  VALUE SPACES.
029                05 SQFT-A3C             PIC Z(5)9.
029                05 FILLER               PIC X(18)  VALUE SPACES.
029                05 SQFT-NSDA            PIC Z(5)9.
029                05 SQFT-VEH             PIC Z(5)9.
029            WORKING-STORAGE SECTION.
029            77  C-FLAG             PICTURE IS X    VALUE SPACE.
029            77  E-FLAG             PICTURE IS X    VALUE SPACE.
029            77  COUNT              PICTURE IS 9(06).
029            77  TUNEXT             PICTURE IS 9.
029            77  N2REQ              PICTURE IS 9(10)    COMPUTATIONAL.
029            77  N3REQ              PICTURE IS 9(10)    COMPUTATIONAL.
029            77  F5KEY              PICTURE IS 9(06).
029            77  UNCOUNT            PICTURE IS 9(06)    VALUE 18.
029            77  CAT                PICTURE IS 9(02).
029            01  CAT-ARRAYS.
029                02  STONA OCCURS 18 TIMES PIC 9(06).
029                02  SQFTA OCCURS 18 TIMES PIC 9(06).
029            01  HEAD-LINE.
029                05 WD01-05 PIC X(30) VALUE '    SRC        ULC     UTC    PAX'.
029                05 WD06-10 PIC X(30) VALUE '    A0    B0     B1      A1      J1'.
029                05 WD11-14 PIC X(24) VALUE '   B2C   B2D   A2C   A2D'.
029                05 WD15-18 PIC X(24) VALUE '   J2D   J2C   A3C   J3C'.
029                05 WD19-22 PIC X(24) VALUE '   J7C  AMMO  UE/B  AS/A'.
029            01  F5-WORK.
029                05 F5SEQ             PICTURE IS 9(05).
029                05 FILLER            PICTURE IS X(61).
029            01  F5-DESC.
029                05 F5MAXRCRDS         PIC 9(10)  VALUE 35000 COMPUTATIONAL.
029                05 F5RECLEN           PIC 9(10)   VALUE 66    COMPUTATIONAL.
029                05 FILLER             PIC 9(10)   VALUE 0     COMPUTATIONAL.
029                05 F5KEYLEN           PIC 9(10)   VALUE 6     COMPUTATIONAL.
029                05 F5NROFA            PIC 9(10)   VALUE 5     COMPUTATIONAL.
029                05 F5NIOFA            PIC 9(10)   VALUE 20    COMPUTATIONAL.
029                05 F5DWD11.
029                07 F5DWD11A           PIC X(03)   VALUE SPACES.
```

```
029                07 F5DWD11B        PIC X(03)    VALUE SPACES.
029          01    F5-INFO.
029                05 F5RBLKS         PICTURE IS 9(10)    COMPUTATIONAL.
029                05 F5IBLKS         PICTURE IS 9(10)    COMPUTATIONAL.
029                05 F5NIOFB         PICTURE IS 9(10)    COMPUTATIONAL.
029                05 F5NRCRD         PICTURE IS 9(10)    COMPUTATIONAL.
029                05 F5NRIOF         PICTURE IS 9(10)    COMPUTATIONAL.
029                05 F5NDEL          PICTURE IS 9(10)    COMPUTATIONAL.
029                05 F5NREAD         PICTURE IS 9(10)    COMPUTATIONAL.
029                05 F5NFIOF         PICTURE IS 9(10)    COMPUTATIONAL.
029                05 F5NWRITE        PICTURE IS 9(10)    COMPUTATIONAL.
029          01    F5-INFORMATION.
029                05 F5RBLKS         PICTURE IS 9(10).
029                05 FILLER          PICTURE IS X(02)    VALUE IS SPACES.
029                05 F5IBLKS         PICTURE IS 9(10).
029                05 FILLER          PICTURE IS X(02)    VALUE IS SPACES.
029                05 F5NIOFB         PICTURE IS 9(10).
029                05 FILLER          PICTURE IS X(02)    VALUE IS SPACES.
029                05 F5NRCRD         PICTURE IS 9(10).
029                05 FILLER          PICTURE IS X(02)    VALUE IS SPACES.
029                05 F5NRIOF         PICTURE IS 9(10).
029                05 FILLER          PICTURE IS X(02)    VALUE IS SPACES.
029                05 F5NDEL          PICTURE IS 9(10).
029                05 FILLER          PICTURE IS X(02)    VALUE IS SPACES.
029                05 F5NREAD         PICTURE IS 9(10).
029                05 FILLER          PICTURE IS X(02)    VALUE IS SPACES.
029                05 F5NFIOF         PICTURE IS 9(10).
029                05 FILLER          PICTURE IS X(02)    VALUE IS SPACES.
029                05 F5NWRITE        PICTURE IS 9(10).
029                05 FILLER          PICTURE IS X(02)    VALUE IS SPACES.
029          01    CARD-WORK.
029                05 CUT-OFF         PICTURE IS 9(05).
029                05 TCOPY-FLAG      PICTURE IS X(05).
029                88 TCOPY           PICTURE IS X(05)    VALUE IS 'TCOPY'.
029                05 LOAD5-FLAG      PICTURE IS X(05).
029                88 LOAD5           PICTURE IS X(05)    VALUE IS 'LOAD5'.
029                05 PRINT-FLAG1     PICTURE IS X(05).
029                88 PRINT-TY        PICTURE IS X(05)    VALUE IS 'TYPEA'.
029                05 PRINT-FLAG2     PICTURE IS X(05).
029                88 PRINT-F1        PICTURE IS X(05)    VALUE IS 'ABF1I'.
029                05 PRINT-FLAG3     PICTURE IS X(05).
029                88 PRINT-L1        PICTURE IS X(05)    VALUE IS 'ABF1O'.
029                05 PRINT-FLAG4     PICTURE IS X(05).
029                88 PRINT-F2        PICTURE IS X(05)    VALUE IS 'TUF2I'.
029                05 PRINT-FLAG5     PICTURE IS X(05).
029                88 PRINT-L2        PICTURE IS X(05)    VALUE IS 'TUF2O'.
029                05 PRINT-FLAG6     PICTURE IS X(05).
029                88 PRINT-F3        PICTURE IS X(05)    VALUE IS 'TUF3I'.
029                05 PRINT-FLAG7     PICTURE IS X(05).
029                88 PRINT-LO        PICTURE IS X(05)    VALUE IS 'FIF2O'.
029                05 FILLER          PICTURE IS X(30).
029          01    ERROR-WORK.
029                05 NUMB            PICTURE IS X(06).
029                05 FILLER          PICTURE IS X(01)    VALUE IS SPACE.
029                05 MESSAGE         PICTURE IS X(125).
029          PROCEDURE DIVISION.
029          DECLARATIVES.
```

```
029          BAD-TAPE SECTION. USE AFTER STANDARD ERROR PROCEDURE ON TUCHA.
029          TUCHA-ERROR-PROC.
029              DISPLAY 'TAPE ERROR ON TUCHA'.
029              DISPLAY 'TYPE LOST  -  PREVIOUS LOG REC'.
029              MOVE TUNEXT TO NUMB.
029              MOVE LOG-REC TO MESSAGE.
029              DISPLAY ERROR-WORK.
029              MOVE 'E' TO E-FLAG.
029              IF TUNEXT EQUALS 1 MOVE 'C' TO C-FLAG.
029          END DECLARATIVES.
029          NORMAL-PROCESSING-SECTION.
029          OPEN-FILES.
029              OPEN INPUT TUCHA CARD-FILE  OUTPUT PRINT-FILE.
029              IF TCOPY OPEN OUTPUT SAVE-FILE.
029              IF LOAD5 OPEN OUTPUT F5.
029          READ-CARD.
029              READ CARD-FILE INTO CARD-WORK AT END GO TO END-CARD.
029              READ CARD-FILE INTO CARD-WORK AT END GO TO END-CARD.
029          END-CARD.
029              CLOSE CARD-FILE.
029              MOVE ZERO TO COUNT.
029              MOVE ZERO TO F5KEY.
029              SORT SORT-FILE ON ASCENDING KEY SRC IN LOG-REC
029                UTC IN LOG-REC CDTYPE CRCAT CONCO IN LOG-REC
029              INPUT  PROCEDURE IS NEXT-TUCHA THRU END-TUCHA
029              OUTPUT PROCEDURE IS LOAD-F5    THRU END-LOAD.
029              GO TO CLOSE-FILES.
029          NEXT-TUCHA.
029              IF COUNT IS NOT LESS THAN CUT-OFF GO TO CLOSE-TUCHA.
029              READ TUCHA  AT END GO TO CLOSE-TUCHA.
029              IF E-FLAG EQUALS 'E' MOVE SPACE TO E-FLAG GO TO NEXT-TUCHA.
029              IF TCOPY WRITE SAVE-REC FROM F1-REC.
029              IF TUTYPE EQUALS 1  IF TYPEA-RCRD  GO TO USE-TYPEA
029                                  ELSE           GO TO USE-ABF1
029              ELSE  IF F2TYPE                    GO TO USE-TUF2
029              ELSE  IF F3TYPE                    GO TO USE-TUF3
029              ELSE                               GO TO TYPE-ERR.
029          TYPE-ERR.
029              MOVE TUTYPE TO NUMB.
029              MOVE 'TYPE TUCHA RECORD NOT EXPECTED' TO MESSAGE.
029              WRITE PRINT-LINE FROM ERROR-WORK.
029              GO TO CLOSE-TUCHA.
029          USE-TYPEA.
029              IF PRINT-TY MOVE TYPEA TO PRINT-LINE PERFORM PRINT-OUT.
029               GO TO NEXT-TUCHA.
029          USE-ABF1.
029              IF PRINT-F1
029                  MOVE SRC IN F1-REC TO PRINT-LINE PERFORM PRINT-OUT
029                  MOVE ABF1          TO PRINT-LINE PERFORM PRINT-OUT.
029              IF CANCELLED MOVE 'C' TO C-FLAG GO TO NEXT-TUCHA
029              ELSE MOVE SPACE TO C-FLAG.
029              ADD 1 TO COUNT.
029              MOVE SPACES TO LOG-REC.
029              MOVE COUNT TO SEQNO IN LOG-REC.
029              MOVE 'F1'  TO CDTYPE IN LOG-REC.
029              MOVE NOPAX TO PERTO IN LOG-REC.
029              MOVE ULC IN F1-REC  TO LEVCO.
```

```
029            MOVE AUTH-WAR-STR TO UNSTR.
029            MOVE SHORT-NAME TO UNAME.
029            MOVE DEPLOYMENT TO DPIND.
029            MOVE SRC IN F1-REC TO SRC IN LOG-REC.
029            MOVE UTC IN ABF1 TO UTC IN LOG-REC.
029            MOVE UIC IN ABF1 TO UIC IN LOG-REC.
029            IF PRINT-L1 MOVE LOG-REC TO PRINT-LINE PERFORM PRINT-OUT.
029            IF SRCID EQUALS '1' RELEASE LOG-REC.
029            MOVE NOF2 TO N2REQ.
029            IF N2REQ EQUALS ZERO MOVE 1 TO TUNEXT
029            ELSE MOVE 2 TO TUNEXT.
029            GO TO NEXT-TUCHA.
029        USE-TUF2.
029            IF PRINT-F2 MOVE TUF2 TO PRINT-LINE PERFORM PRINT-OUT.
029            IF C-FLAG EQUALS 'C' GO TO NEXT-TUCHA.
029            ADD 1 TO COUNT.
029            MOVE COUNT TO SEQNO IN LOG-REC.
029            MOVE 'F2'  TO CDTYPE IN LOG-REC.
029            MOVE ZEROS TO PERTO IN LOG-REC.
029            MOVE CARGO TO CRCAT IN LOG-REC.
029            MOVE CONTAINER TO CONCO IN LOG-REC.
029            MOVE F2SQFT TO SQFT IN LOG-REC.
029            MOVE ZERO   TO STOF IN LOG-REC.
029            MOVE F2STON TO STON IN LOG-REC.
029            MOVE  ZERO              TO MTOF  IN LOG-REC.
029            IF BULKPOL MOVE HBPOL  TO WHMTO IN LOG-REC
029            ELSE       MOVE F2MTON TO WHMTO IN LOG-REC.
029            MOVE  ZERO              TO TEMET IN LOG-REC.
029            MOVE HLC IN TUF2 TO HELCO IN LOG-REC.
029            IF PRINT-L2 MOVE LOG-REC TO PRINT-LINE PERFORM PRINT-OUT.
029            MOVE UTC IN ABF1 TO UTC IN LOG-REC.
029            IF SRCID EQUALS '1' RELEASE LOG-REC.
029            SUBTRACT 1 FROM N2REQ.
029            IF N2REQ EQUALS ZERO MOVE 1 TO TUNEXT
029            ELSE MOVE NOF3 TO N3REQ
029                 IF N3REQ EQUALS ZERO MOVE 2 TO TUNEXT
029                 ELSE MOVE 3 TO TUNEXT.
029            GO TO NEXT-TUCHA.
029        USE-TUF3.
029            IF PRINT-F3 MOVE TUF3  TO PRINT-LINE PERFORM PRINT-OUT.
029            SUBTRACT 1 FROM N3REQ.
029            IF N3REQ EQUALS ZERO MOVE 2 TO TUNEXT.
029            GO TO NEXT-TUCHA.
029        CLOSE-TUCHA.
029            CLOSE TUCHA .
029            IF TCOPY CLOSE SAVE-FILE.
029        END-TUCHA.
029            EXIT.
029        LOAD-F5.
029            RETURN SORT-FILE RECORD INTO F5-WORK AT END GO TO LAST-SUM.
029            ADD 1 TO F5KEY.
029            MOVE F5KEY TO F5SEQ IN F5-WORK.
029            IF PRINT-LO MOVE F5-WORK TO PRINT-LINE PERFORM PRINT-OUT.
029            IF LOAD5 WRITE RF5 FROM F5-WORK  INVALID KEY GO TO KEY-ERR.
029            IF CDTYPE EQUALS 'F1' GO TO F1ROLL ELSE GO TO F2ROLL.
029        F2ROLL.
032            IF CRCAT1 EQUALS 'A' OR 'C' GO TO VEHICLE.
```

```
029            IF CRCAT1 EQUALS 'B' GO TO NSDA.
029            IF CRCAT1 EQUALS 'J' GO TO NON-VEH.
029            IF CRCAT1 EQUALS 'M' GO TO AMMO.
029            MOVE CRCAT TO NUMB.
029            MOVE 'CRCAT NOT EXPECTED' TO MESSAGE.
029            WRITE PRINT-LINE FROM ERROR-WORK.
029            GO TO LOAD-FS.
029        VEHICLE.
029            IF CONCO EQUALS 'M' OR 'C' MOVE 8 TO CAT
029            ELSE MOVE 9 TO CAT.
029            IF CRCAT2 EQUALS '0' MOVE 1  TO CAT.
029            IF CRCAT2 EQUALS '1' MOVE 4  TO CAT.
029            IF CRCAT2 EQUALS '3' MOVE 12 TO CAT.
029            ADD SQFT TO SQFTA(17).
029            GO TO SUM-CAT.
029        NSDA.
029            IF CONCO EQUALS 'C' OR 'B' MOVE 6 TO CAT
029            ELSE MOVE 7 TO CAT.
029            IF CRCAT2 EQUALS '0' MOVE 2  TO CAT.
029            IF CRCAT2 EQUALS '1' MOVE 3  TO CAT.
029            ADD SQFT TO SQFTA(16).
029            GO TO SUM-CAT.
029        NON-VEH.
029            IF CONCO EQUALS 'C' OR 'B' MOVE 11 TO CAT
029            ELSE MOVE 10 TO CAT.
029            IF CRCAT2 EQUALS '0' MOVE 1  TO CAT.
029            IF CRCAT2 EQUALS '1' MOVE 5  TO CAT.
029            IF CRCAT2 EQUALS '3' MOVE 13 TO CAT.
029            IF CRCAT2 EQUALS '7' MOVE 14 TO CAT.
029            GO TO SUM-CAT.
029        AMMO.
029            MOVE 15 TO CAT.
029        SUM-CAT.
029            ADD STON TO STONA(CAT).
029            ADD SQFT TO SQFTA(CAT).
029            IF CAT IS LESS THAN 14 ADD STON TO STONA(16)
029            ELSE                   ADD STON TO STONA(17).
029            GO TO LOAD-FS.
029        FIROLL.
029            IF FSSEQ EQUALS 1 GO TO FISAVE.
029        PRINT-SUM.
029            IF UNCOUNT EQUALS 18 MOVE ZERO TO UNCOUNT
029            WRITE PRINT-LINE FROM HEAD-LINE
029            AFTER ADVANCING TOP-OF-PAGE LINES.
029            ADD 1 TO UNCOUNT.
029            MOVE STONA(1)          TO STON-A0  IN LINE1.
029            MOVE STONA(2)          TO STON-B0  IN LINE1.
029            MOVE STONA(3)          TO STON-B1  IN LINE1.
029            MOVE STONA(4)          TO STON-A1  IN LINE1.
029            MOVE STONA(5)          TO STON-J1  IN LINE1.
029            MOVE STONA(6)          TO STON-B2C IN LINE1.
029            MOVE STONA(7)          TO STON-B2D IN LINE1.
029            MOVE STONA(8)          TO STON-A2C IN LINE1.
029            MOVE STONA(9)          TO STON-A2D IN LINE1.
029            MOVE STONA(10)         TO STON-J2D IN LINE1.
029            MOVE STONA(11)         TO STON-J2C IN LINE1.
029            MOVE STONA(12)         TO STON-A3C IN LINE1.
```

```
029          MOVE STONA(13)       TO STON-J3C IN LINE1.
029          MOVE STONA(14)       TO STON-J7C IN LINE1.
029          MOVE STONA(15)       TO STON-M7C IN LINE1.
029          MOVE STONA(16)       TO STON-UE  IN LINE1.
029          MOVE STONA(17)       TO STON-AS  IN LINE1.
029          WRITE PRINT-LINE FROM LINE1 AFTER ADVANCING 2 LINES.
029          MOVE SQFTA(1)        TO SQFT-A0  IN LINE2.
029          MOVE SQFTA(2)        TO SQFT-B0  IN LINE2.
029          MOVE SQFTA(3)        TO SQFT-B1  IN LINE2.
029          MOVE SQFTA(4)        TO SQFT-A1  IN LINE2.
029          MOVE SQFTA(6)        TO SQFT-B2C IN LINE2.
029          MOVE SQFTA(7)        TO SQFT-B2D IN LINE2.
029          MOVE SQFTA(8)        TO SQFT-A2C IN LINE2.
029          MOVE SQFTA(9)        TO SQFT-A2D IN LINE2.
029          MOVE SQFTA(12)       TO SQFT-A3C IN LINE2.
029          MOVE SQFTA(16)       TO SQFT-NSDA.
029          MOVE SQFTA(17)       TO SQFT-VEH IN LINE2.
029          WRITE PRINT-LINE FROM LINE2.
029          ENTER FORTRAN SDFTU SUBROUTINE.
029      FISAVE.
029          MOVE SRC19  IN LOG-REC TO SRC1-9 IN LINE1.
029          MOVE SRC11-12          TO SRCPAR IN LINE1.
029          MOVE LEVCO             TO ULC    IN LINE1.
029          MOVE UTC    IN LOG-REC TO UTC    IN LINE1.
029          MOVE UNSTR             TO STR    IN LINE1.
029          MOVE SRC1   IN LINE1   TO SRC2   IN LINE2.
029          MOVE UNAME             TO NAME   IN LINE2.
029          MOVE PERTO IN LOG-REC  TO PAX    IN LINE2.
029          PERFORM ZERO-ARRAYS VARYING CAT FROM 1 BY 1
029          UNTIL CAT IS GREATER THAN 18.
029          GO TO LOAD-F5.
029      ZERO-ARRAYS.
029          MOVE ZEROS TO STONA(CAT) SQFTA(CAT).
029      LAST-SUM.
029          PERFORM PRINT-SUM.
029          ENTER FORTRAN EOF12 SUBROUTINE.
029      END-LOAD.
029          EXIT.
029      KEY-ERR.
029          MOVE F5KEY TO NUMB.
029          MOVE 'INVALID KEY FOR FIF2 FILE' TO MESSAGE.
029          MOVE ERROR-WORK TO PRINT-LINE.
029      PRINT-OUT.
029          WRITE PRINT-LINE.
029      CLOSE-FILES.
029          IF LOAD5  CLOSE F5 USING F5-INFO.
029          MOVE 'F5 INFORMATION AREA' TO PRINT-LINE.
029          PERFORM PRINT-OUT.
029          MOVE CORRESPONDING F5-INFO TO F5-INFORMATION.
029          MOVE  F5-INFORMATION       TO PRINT-LINE.
029          PERFORM PRINT-OUT.
029          CLOSE PRINT-FILE.
029          STOP RUN.

END ELT.
```

```
@ELT,L    32MARTIN.SDFTU
ELT007 RL1870 08/16-07:41:01-(1,)
        000              SUBROUTINE SDFTU
        000              COMMON DUMMY, LINE1(22), LINE2(22)
        000              WRITE (12,4) LINE1
        000              WRITE (12,4) LINE2
        000              RETURN
        000            4 FORMAT (22A6)
        000              ENTRY EOF12
        000              ENDFILE  12
        000              RETURN
        000              END

END ELT.

@HDG     MRG   PRE-PROCESSOR PROGRAM POMCUSROLL        ••UNCLASSIFIED•• •L,0
```

```
@ELT,L   32MARTIN.POMSUM
ELT007 RLIB70 08/16-07:41:03-(19,)
        015          IDENTIFICATION DIVISION.
        015          PROGRAM-ID.        LOAD-FIF2.
        015          AUTHOR.            G.L.MARTIN
        015          DATE-WRITTEN.      JANUARY 1975
        015          DATE-COMPILED.
        015          INSTALLATION.      US ARMY CAA.
        015          SECURITY.          UNCLASSIFIED.
        015          REMARKS.           PROGRAM LOADS CAMP FIF2 FILE FROM A
        015              NONSTANDARD POMCUS FILE WITH SRC APPENDED TO THE
        015              TUF1 RECORD.
        015          ENVIRONMENT DIVISION.
        015          CONFIGURATION SECTION.
        015          SOURCE-COMPUTER. UNIVAC-1108.
        015          OBJECT-COMPUTER. UNIVAC-1108.
        015          INPUT-OUTPUT SECTION.
        015          FILE-CONTROL.
        015              SELECT PRINT-FILE ASSIGN TO PRINTER.
        015              SELECT CARD-FILE ASSIGN TO CARD-READER.
        015              SELECT POMCUS ASSIGN TO UNISERVO POMCUS.
        015              SELECT SAVE-FILE ASSIGN TO UNISERVO SAVE.
        015              SELECT SORT-FILE ASSIGN TO MASS-STORAGE XA.
        015              SELECT F5 ASSIGN TO MASS-STORAGE 32FIF2
        015                       ORGANIZATION IS INDEXED
        015                       FILE-DESCRIPTION IS F5-DESC
        015                       ACCESS MODE IS RANDOM
        015                       SYMBOLIC KEY IS F5KEY.
        015          DATA DIVISION.
        015          FILE SECTION.
        015          FD  PRINT-FILE
        015              LABEL RECORDS ARE OMITTED
        015              DATA RECORD IS PRINT-LINE.
        015          01  PRINT-LINE              PICTURE IS X(132).
        015          FD  CARD-FILE
        015              LABEL RECORDS ARE OMITTED
        015              DATA RECORD IS INPUT-CARD.
        015          01  INPUT-CARD              PICTURE IS X(80).
        015          FD  SAVE-FILE
        015              LABEL RECORDS ARE OMITTED
        015              BLOCK CONTAINS 24 RECORDS
        015              RECORD CONTAINS 138 CHARACTERS
        015              DATA RECORD IS SAVE-REC.
        015          01  SAVE-REC                PICTURE IS X(138).
        015          FD  POMCUS
        015              LABEL RECORDS ARE OMITTED
        015              RECORDING MODE IS 1
        015              BLOCK CONTAINS 10 RECORDS
        015              RECORD CONTAINS 120 CHARACTERS
        015              DATA RECORDS ARE F1-REC, F2-REC, F3-REC.
        015          01  F1-REC.
        015              02 TUF1.
        015              03 TYPEA.
        015              05 FILLER               PICTURE IS X(06).
        015              05 TUTYPE               PICTURE IS 9(01).
        015              88 F1TYPE               PICTURE IS 9(01)    VALUE IS 1.
        015              88 F2TYPE               PICTURE IS 9(01)    VALUE IS 2.
```

```
015                88 F3TYPE          PICTURE IS 9(01)   VALUE IS 3.
015                05 FILLER          PICTURE IS X(01).
015                05 UTC             PICTURE IS X(05).
015                05 POMCUS-CODE1    PICTURE IS X(01).
015                05 FILLER          PICTURE IS X(04).
015                05 NOPAX           PICTURE IS 9(05).
015                05 F2REQ           PICTURE IS 9(03).
015                05 FILLER          PICTURE IS X(44).
015                05 UIC             PICTURE IS X(06).
015                05 FILLER          PICTURE IS X(13).
015                05 SRC             PICTURE IS X(12).
015                05 FILLER          PICTURE IS X(19).
015                66 AUTH-WAR-STR RENAMES NOPAX.
015                66 NOF2          RENAMES F2REQ.
015             01 F2-REC.
015                03 TUF2.
015                05 FILLER          PICTURE IS X(06).
015                05 TUTYPE2         PICTURE IS X(01).
015                05 FILLER          PICTURE IS X(01).
015                05 UTC             PICTURE IS X(05).
015                05 POMCUS-CODE2    PICTURE IS A(01).
015                88 TAT             PICTURE IS A(01)  VALUE IS 'A'.
015                88 OTHER           PICTURE IS A(01)  VALUE IS 'B'.
015                88 PREPO           PICTURE IS A(01)  VALUE IS 'C'.
015                05 FILLER          PICTURE IS X(19).
015                05 CARGO.
015                07 CARGO1          PICTURE IS A(01).
015                88 BULKPOL         PICTURE IS A(01)   VALUE IS 'G'.
015                07 CARGO2          PICTURE IS 9(01).
015                05 CONTAINER       PICTURE IS A(01).
015                05 F2SQFT          PICTURE IS 9(06).
015                05 FILLER          PICTURE IS X(02).
015                05 F2STON          PICTURE IS 9(06).
015                05 FILLER          PICTURE IS X(01).
015                05 F2MTON          PICTURE IS 9(06).
015                05 FILLER          PICTURE IS X(01).
015                05 HLC             PICTURE IS A(01).
015                05 F3REQ           PICTURE IS 9(03).
015                05 FILLER          PICTURE IS X(08).
015                05 F2UIC           PICTURE IS X(06).
015                05 FILLER          PICTURE IS X(13).
015                05 F2SRC           PICTURE IS X(12).
015                05 FILLER          PICTURE IS X(19).
015                66 NOF3  RENAMES   F3REQ.
015             01 F3-REC.
015                03 TUF3.
015                05 FILLER          PICTURE IS X(06).
015                05 TUTYPE3         PICTURE IS 9(01).
015                05 FILLER          PICTURE IS X(01).
015                05 UTC3            PICTURE IS X(05).
015                05 POMCUS-CODE3    PICTURE IS A(01).
015                05 FILLER          PICTURE IS X(04).
015                05 CARGO3          PICTURE IS X(03).
015                05 F3IDNO          PICTURE IS 9(03).
015                05 F3DESC          PICTURE IS X(13).
015                05 LENGTH          PICTURE IS 9(04).
015                05 WIDTH           PICTURE IS 9(03).
```

```
000113    015          05 HEIGHT           PICTURE IS 9(03).
000114    015          05 F3SQFT           PICTURE IS 9(04).
000115    015          05 PIECES           PICTURE IS 9(03).
000116    015          05 F3STON           PICTURE IS 9(06).
000117    015          05 F3MTON           PICTURE IS 9(06).
000118    015          05 FILLER           PICTURE IS X(04).
000119    015          05 F3UIC            PICTURE IS X(06).
000120    015          05 FILLER           PICTURE IS X(13).
000121    015          05 F3SRC            PICTURE IS X(12).
000122    015          05 FILLER           PICTURE IS X(19).
000123    016      SD  SORT-FILE           DATA RECORD IS SORT-REC.
000124    016      01  SORT-REC.
000125    016          05 FILLER           PICTURE IS X(05).
000126    017          05 SRTYPE           PICTURE IS X(02).
000127    016          05 FILLER           PICTURE IS X(05).
000128    016          05 SRCAT            PICTURE IS X(03).
000129    016          05 FILLER           PICTURE IS X(40).
000130    016          05 SRUIC            PICTURE IS X(06).
000131    015          05 FILLER           PICTURE IS X(01).
000132    016          05 SRPOM            PICTURE IS A(01).
000133    015          05 FILLER           PICTURE IS X(03).
000134    015      FD  F5
000135    015          LABEL RECORDS ARE STANDARD
000136    015          DATA RECORD IS RF5.
000137    015      01  RF5                 PICTURE IS X(5376).
000138    015      COMMON-STORAGE SECTION.
000139    015      01  LINE1.
000140    015          05 SRC1.
000141    015          07 SRC1-9                 PIC X(9).
000142    015          07 SRCPAR                 PIC X(2).
000143    015          05 LINE-1                 PIC X(1)    VALUE '1'.
000144    015          05 FILLER                 PIC X(2)    VALUE SPACES.
000145    015          05 ULC                    PIC X(3).
000146    015          05 FILLER                 PIC X(2)    VALUE SPACES.
000147    015          05 UTC                    PIC X(5).
000148    015          05 STR                    PIC Z(5)9.
000149    015          05 STON-A0                PIC Z(5)9.
000150    015          05 STON-B0                PIC Z(5)9.
000151    015          05 STON-B1                PIC Z(5)9.
000152    015          05 STON-A1                PIC Z(5)9.
000153    015          05 STON-J1                PIC Z(5)9.
000154    015          05 STON-B2C               PIC Z(5)9.
000155    015          05 STON-B2D               PIC Z(5)9.
000156    015          05 STON-A2C               PIC Z(5)9.
000157    015          05 STON-A2D               PIC Z(5)9.
000158    015          05 STON-J2D               PIC Z(5)9.
000159    015          05 STON-J2C               PIC Z(5)9.
000160    015          05 STON-A3C               PIC Z(5)9.
000161    015          05 STON-J3C               PIC Z(5)9.
000162    015          05 STON-J7C               PIC Z(5)9.
000163    015          05 STON-M7C               PIC Z(5)9.
000164    015          05 STON-UE                PIC Z(5)9.
000165    015          05 STON-AS                PIC Z(5)9.
000166    015      01  LINE2.
000167    015          05 UIC                    PIC X(06).
000168    015          05 FILLER                 PIC X(05)    VALUE SPACES.
000169    015          05 LINE-2                 PIC X(1)    VALUE '2'.
```

```
015          05 NAME              PIC X(12).
015          05 PAX               PIC Z(5)9.
015          05 SQFT-A0           PIC Z(5)9.
015          05 SQFT-B0           PIC Z(5)9.
015          05 SQFT-B1           PIC Z(5)9.
015          05 SQFT-A1           PIC Z(5)9.
015          05 FILLER            PIC X(6)   VALUE SPACES.
015          05 SQFT-B2C          PIC Z(5)9.
015          05 SQFT-B2D          PIC Z(5)9.
015          05 SQFT-A2C          PIC Z(5)9.
015          05 SQFT-A2D          PIC Z(5)9.
015          05 FILLER            PIC X(12)  VALUE SPACES.
015          05 SQFT-A3C          PIC Z(5)9.
015          05 FILLER            PIC X(18)  VALUE SPACES.
015          05 SQFT-NSDA         PIC Z(5)9.
015          05 SQFT-VEH          PIC Z(5)9.
015       WORKING-STORAGE SECTION.
015       77  C-FLAG          PICTURE IS X    VALUE SPACE.
015       77  E-FLAG          PICTURE IS X    VALUE SPACE.
015       77  COUNT           PICTURE IS 9(06).
015       77  TUNEXT          PICTURE IS 9.
015       77  N2REQ           PICTURE IS 9(10)    COMPUTATIONAL.
015       77  N3REQ           PICTURE IS 9(10)    COMPUTATIONAL.
015       77  F5KEY           PICTURE IS 9(06).
015       77  UNCOUNT         PICTURE IS 9(06)    VALUE 10.
015       77  PT              PICTURE IS 9(01)    VALUE 1.
015       77  CAT             PICTURE IS 9(02).
015       77  UIC-SEQ         PICTURE IS 9(02).
015       77  NO-UICS         PICTURE IS 9(02)  VALUE ZERO.
015       01  UIC-ARRAY.
015           03  UIC-TABLE OCCURS 100 TIMES.
015           05  OLD-UIC     PICTURE IS X(06).
015           05  NEW-UIC     PICTURE IS X(06).
015       01  UIC-CARD.
015           03  NOT-UIC     PICTURE IS X(06).
015           03  FILLER      PICTURE IS X(01).
015           03  POM-UIC     PICTURE IS X(06).
015       01  CAT-ARRAYS.
015           03  POMCUS-TYPE OCCURS 2 TIMES.
015           05  STONA OCCURS 18 TIMES PIC 9(06).
015           05  SQFTA OCCURS 18 TIMES PIC 9(06).
016       01  LOG-REC.
016           05 SEQNO        PICTURE IS 9(05).
016           05 CDTYPE.
016           07 CDTYPE1      PICTURE IS A(01).
016           07 CDTYPE2      PICTURE IS 9(01).
016           05 PERTO        PICTURE IS 9(05).
016           05 CRCAT.
016           07 CRCAT1       PICTURE IS X(01).
016           07 CRCAT2       PICTURE IS X(01).
016           05 CONCO        PICTURE IS X(01).
016           05 SQFT         PICTURE IS 9(06).
016           05 STOF         PICTURE IS 9(02).
016           05 STON         PICTURE IS 9(06).
016           05 MTON.
016           07 MTOF         PICTURE IS 9(01).
016           07 WHMTO        PICTURE IS 9(06).
```

```
016              07 TEMET            PICTURE IS 9(01).
016              05 HELCO            PICTURE IS A(01).
016              05 SRC.
016              07 SRC19            PICTURE IS X(09).
016              07 SRC10            PICTURE IS X(01).
016              07 SRC11-12         PICTURE IS X(02).
016              05 UTC              PICTURE IS X(05).
016              05 UIC              PICTURE IS X(06).
016              05 FILLER           PICTURE IS X(01).
016              05 POMCUS-CODE      PICTURE IS A(01).
016              05 FILLER           PICTURE IS X(03).
016              66 LEVCO  RENAMES CRCAT THRU CONCO.
016              66 UNSTR  RENAMES SQFT.
016              66 UNAME  RENAMES STOF  THRU TEMET.
016              66 DPIND  RENAMES HELCO.
015          01  HEAD-LINE.
015              05 WD01-05 PIC X(30) VALUE '    SRC        ULC     UTC    PAX'.
015              05 WD06-10 PIC X(30) VALUE '    A0     B0     B1     A1     J1'.
015              05 WD11-14 PIC X(24) VALUE '  B2C   B2D   A2C   A2D'.
015              05 WD15-18 PIC X(24) VALUE '  J2D   J2C   A3C   J3C'.
015              05 WD19-22 PIC X(24) VALUE '  J7C  AMMO  UE/B  AS/A'.
015          01  F5-WORK.
015              05 F5SEQ           PICTURE IS 9(05).
015              05 FILLER          PICTURE IS X(61).
015          01  F5-DESC.
015              05 F5MAXRCRDS       PIC 9(10)  VALUE 35000 COMPUTATIONAL.
015              05 F5RECLEN         PIC 9(10)   VALUE 66    COMPUTATIONAL.
015              05 FILLER           PIC 9(10)   VALUE 0     COMPUTATIONAL.
015              05 F5KEYLEN         PIC 9(10)   VALUE 6     COMPUTATIONAL.
015              05 F5NROFA          PIC 9(10)   VALUE 5     COMPUTATIONAL.
015              05 F5NIOFA          PIC 9(10)   VALUE 20    COMPUTATIONAL.
015              05 F5DWD11.
015              07 F5DWD11A         PIC X(03)   VALUE SPACES.
015              07 F5DWD11B         PIC X(03)   VALUE SPACES.
015          01  F5-INFO.
015              05 F5RBLKS          PICTURE IS 9(10)   COMPUTATIONAL.
015              05 F5IBLKS          PICTURE IS 9(10)   COMPUTATIONAL.
015              05 F5NIOFB          PICTURE IS 9(10)   COMPUTATIONAL.
015              05 F5NRCRD          PICTURE IS 9(10)   COMPUTATIONAL.
015              05 F5NRIOF          PICTURE IS 9(10)   COMPUTATIONAL.
015              05 F5NDEL           PICTURE IS 9(10)   COMPUTATIONAL.
015              05 F5NREAD          PICTURE IS 9(10)   COMPUTATIONAL.
015              05 F5NFIOF          PICTURE IS 9(10)   COMPUTATIONAL.
015              05 F5NWRITE         PICTURE IS 9(10)   COMPUTATIONAL.
015          01  F5-INFORMATION.
015              05 F5RBLKS          PICTURE IS 9(10).
015              05 FILLER           PICTURE IS X(02)   VALUE IS SPACES.
015              05 F5IBLKS          PICTURE IS 9(10).
015              05 FILLER           PICTURE IS X(02)   VALUE IS SPACES.
015              05 F5NIOFB          PICTURE IS 9(10).
015              05 FILLER           PICTURE IS X(02)   VALUE IS SPACES.
015              05 F5NRCRD          PICTURE IS 9(10).
015              05 FILLER           PICTURE IS X(02)   VALUE IS SPACES.
015              05 F5NRIOF          PICTURE IS 9(10).
015              05 FILLER           PICTURE IS X(02)   VALUE IS SPACES.
015              05 F5NDEL           PICTURE IS 9(10).
015              05 FILLER           PICTURE IS X(02)   VALUE IS SPACES.
```

```
015            05 F5NREAD          PICTURE IS 9(10).
015            05 FILLER           PICTURE IS X(02)   VALUE IS SPACES.
015            05 F5NFIOF          PICTURE IS 9(10).
015            05 FILLER           PICTURE IS X(02)   VALUE IS SPACES.
015            05 F5NWRITE         PICTURE IS 9(10).
015            05 FILLER           PICTURE IS X(02)   VALUE IS SPACES.
015        01  CARD-WORK.
015            05 CUT-OFF          PICTURE IS 9(05).
015            05 TCOPY-FLAG       PICTURE IS X(05).
015            88 TCOPY            PICTURE IS X(05)   VALUE IS 'TCOPY'.
015            05 LOAD5-FLAG       PICTURE IS X(05).
015            88 LOAD5            PICTURE IS X(05)   VALUE IS 'LOAD5'.
015            05 PRINT-FLAG1      PICTURE IS X(05).
015            88 PRINT-TY         PICTURE IS X(05)   VALUE IS 'TYPEA'.
015            05 PRINT-FLAG2      PICTURE IS X(05).
015            88 PRINT-F1         PICTURE IS X(05)   VALUE IS 'TUF11'.
015            05 PRINT-FLAG3      PICTURE IS X(05).
015            88 PRINT-L1         PICTURE IS X(05)   VALUE IS 'TUF10'.
015            05 PRINT-FLAG4      PICTURE IS X(05).
015            88 PRINT-F2         PICTURE IS X(05)   VALUE IS 'TUF21'.
015            05 PRINT-FLAG5      PICTURE IS X(05).
015            88 PRINT-L2         PICTURE IS X(05)   VALUE IS 'TUF20'.
015            05 PRINT-FLAG6      PICTURE IS X(05).
015            88 PRINT-F3         PICTURE IS X(05)   VALUE IS 'TUF31'.
015            05 PRINT-FLAG7      PICTURE IS X(05).
015            88 PRINT-LO         PICTURE IS X(05)   VALUE IS 'FIF20'.
015            05 FILLER           PICTURE IS X(30).
015        01  ERROR-WORK.
015            05 NUMB             PICTURE IS X(06).
015            05 FILLER           PICTURE IS X(01)   VALUE IS SPACE.
015            05 MESSAGE          PICTURE IS X(125).
015        PROCEDURE DIVISION.
015        OPEN-FILES.
015            OPEN INPUT POMCUS CARD-FILE  OUTPUT PRINT-FILE.
015            IF TCOPY OPEN OUTPUT SAVE-FILE.
015            IF LOAD5 OPEN OUTPUT F5.
015        READ-CARD.
015            READ CARD-FILE INTO CARD-WORK AT END GO TO END-CARD.
015            READ CARD-FILE INTO CARD-WORK AT END GO TO END-CARD.
015            MOVE CARD-WORK TO PRINT-LINE  PERFORM PRINT-OUT.
015        NEXT-CARD.
015            ADD 1 TO NO-UICS.
015            READ CARD-FILE INTO UIC-CARD  AT END GO TO END-CARD.
015            MOVE UIC-CARD  TO PRINT-LINE  PERFORM PRINT-OUT.
015            MOVE POM-UIC TO OLD-UIC(NO-UICS).
015            MOVE NOT-UIC TO NEW-UIC(NO-UICS).
015            GO TO NEXT-CARD.
015        END-CARD.
015            CLOSE CARD-FILE.
015            MOVE ZERO TO COUNT.
015            MOVE ZERO TO F5KEY.
016        SORT-CONTROL.
016            SORT SORT-FILE ON ASCENDING KEY SRUIC SRPOM SRTYPE SRCAT
015            INPUT  PROCEDURE IS NEXT-POMCUS THRU END-POMCUS
015            OUTPUT PROCEDURE IS LOAD-F5    THRU END-LOAD.
015            GO TO CLOSE-FILES.
015        NEXT-POMCUS.
```

```
015             IF COUNT IS NOT LESS THAN CUT-OFF GO TO CLOSE-POMCUS.
015             READ POMCUS  AT END GO TO CLOSE-POMCUS.
015             IF E-FLAG EQUALS 'E' MOVE SPACE TO E-FLAG GO TO NEXT-POMCUS.
015             IF TCOPY WRITE SAVE-REC FROM F1-REC.
015                  IF F1TYPE                      GO TO USE-TUF1
015             ELSE IF F2TYPE                      GO TO USE-TUF2
015             ELSE IF F3TYPE                      GO TO USE-TUF3
015             ELSE                                GO TO TYPE-ERR.
015         TYPE-ERR.
015             MOVE TUTYPE TO NUMB.
015             MOVE 'TYPE POMCUS RECORD NOT EXPECTED' TO MESSAGE.
015             WRITE PRINT-LINE FROM ERROR-WORK.
015             GO TO CLOSE-POMCUS.
015         USE-TUF1.
015             IF PRINT-F1
015                MOVE SRC IN F1-REC TO PRINT-LINE PERFORM PRINT-OUT
015                MOVE TUF1          TO PRINT-LINE PERFORM PRINT-OUT.
015             ADD 1 TO COUNT.
015             MOVE SPACES TO LOG-REC.
015             MOVE COUNT TO SEQNO IN LOG-REC.
015             MOVE 'F1'  TO CDTYPE IN LOG-REC.
015             MOVE NOPAX TO PERTO IN LOG-REC.
015             MOVE AUTH-WAR-STR TO UNSTR.
015             MOVE SRC IN F1-REC TO SRC IN LOG-REC.
015             MOVE ZEROS TO SRC11-12.
015             MOVE UTC IN TUF1 TO UTC IN LOG-REC.
015             MOVE UIC IN TUF1 TO UIC IN LOG-REC.
015             MOVE POMCUS-CODE1   TO POMCUS-CODE.
015             IF PRINT-L1 MOVE LOG-REC TO PRINT-LINE PERFORM PRINT-OUT.
016             RELEASE SORT-REC FROM LOG-REC.
015             PERFORM UIC-SEARCH VARYING UIC-SEQ FROM 1 BY 1
015             UNTIL UIC-SEQ EQUALS NO-UICS.
015             MOVE NOF2 TO N2REQ.
015             IF N2REQ EQUALS ZERO MOVE 1 TO TUNEXT
015             ELSE MOVE 2 TO TUNEXT.
015             GO TO NEXT-POMCUS.
015         USE-TUF2.
015             IF PRINT-F2 MOVE TUF2 TO PRINT-LINE PERFORM PRINT-OUT.
015             ADD 1 TO COUNT.
015             MOVE COUNT TO SEQNO IN LOG-REC.
015             MOVE 'F2'  TO CDTYPE IN LOG-REC.
015             MOVE ZEROS TO PERTO IN LOG-REC.
015             MOVE CARGO TO CRCAT IN LOG-REC.
015             MOVE CONTAINER TO CONCO IN LOG-REC.
015             MOVE F2SQFT TO SQFT IN LOG-REC.
015             MOVE ZERO   TO STOF IN LOG-REC.
015             MOVE F2STON TO  STON IN LOG-REC.
015             MOVE F2MTON TO WHMTO IN LOG-REC.
015             MOVE ZERO   TO TEMET IN LOG-REC.
015             MOVE HLC IN TUF2 TO HELCO IN LOG-REC.
015             MOVE UTC IN TUF1 TO UTC IN LOG-REC.
018             MOVE F2UIC        TO UIC IN LOG-REC.
015             MOVE POMCUS-CODE2    TO    POMCUS-CODE.
015             IF PRINT-L2 MOVE LOG-REC TO PRINT-LINE PERFORM PRINT-OUT.
016             RELEASE SORT-REC FROM LOG-REC.
015             PERFORM UIC-SEARCH VARYING UIC-SEQ FROM 1 BY 1
015             UNTIL UIC-SEQ EQUALS NO-UICS.
```

```
015            SUBTRACT 1 FROM N2REQ.
015            IF N2REQ EQUALS ZERO MOVE 1 TO TUNEXT
015            ELSE MOVE NOF3 TO N3REQ
015                 IF N3REQ EQUALS ZERO MOVE 2 TO TUNEXT
015                 ELSE MOVE 3 TO TUNEXT.
015            GO TO NEXT-POMCUS.
015        USE-TUF3.
015            IF PRINT-F3 MOVE TUF3  TO PRINT-LINE PERFORM PRINT-OUT.
015            SUBTRACT 1 FROM N3REQ.
015            IF N3REQ EQUALS ZERO MOVE 2 TO TUNEXT.
015            GO TO NEXT-POMCUS.
015        CLOSE-POMCUS.
015            CLOSE POMCUS .
015            IF TCOPY CLOSE SAVE-FILE.
015        END-POMCUS.
015            EXIT.
015        UIC-SEARCH.
015            IF UIC IN LOG-REC EQUAL OLD-UIC(UIC-SEQ)
016            MOVE NEW-UIC(UIC-SEQ) TO UIC IN LOG-REC
019            RELEASE SORT-REC FROM LOG-REC
019            MOVE OLD-UIC(UIC-SEQ) TO UIC IN LOG-REC.
015        LOAD-F5.
015            RETURN SORT-FILE RECORD INTO F5-WORK AT END GO TO LAST-SUM.
017            MOVE F5-WORK TO LOG-REC.
015            ADD 1 TO F5KEY.
015            MOVE F5KEY TO F5SEQ IN F5-WORK.
015            IF PRINT-LO MOVE F5-WORK TO PRINT-LINE PERFORM PRINT-OUT.
015            IF LOAD5 WRITE RF5 FROM F5-WORK  INVALID KEY GO TO KEY-ERR.
015            IF CDTYPE EQUALS 'F1' GO TO F1ROLL ELSE GO TO F2ROLL.
015        F2ROLL.
015            IF PT EQUALS 1 AND POMCUS-CODE EQUALS 'C'
015            PERFORM PRINT-SUM  MOVE 2 TO PT.
015            IF CRCAT1 EQUALS 'A' OR 'C' GO TO VEHICLE.
015            IF CRCAT1 EQUALS 'B' GO TO NSDA.
015            IF CRCAT1 EQUALS 'J' GO TO NON-VEH.
015            IF CRCAT1 EQUALS 'M' GO TO AMMO.
015            MOVE CRCAT TO NUMB.
015            MOVE 'CRCAT NOT EXPECTED' TO MESSAGE.
015            WRITE PRINT-LINE FROM ERROR-WORK.
015            GO TO LOAD-F5.
015        VEHICLE.
015            IF CONCO EQUALS 'H' OR 'C' MOVE 8 TO CAT
015            ELSE MOVE 9 TO CAT.
015            IF CRCAT2 EQUALS '0' MOVE 1  TO CAT.
015            IF CRCAT2 EQUALS '1' MOVE 4  TO CAT.
015            IF CRCAT2 EQUALS '3' MOVE 12 TO CAT.
015            ADD SQFT TO SQFTA(PT, 17).
015            GO TO SUM-CAT.
015        NSDA.
015            IF CONCO EQUALS 'C' OR 'B' MOVE 6 TO CAT
015            ELSE MOVE 7 TO CAT.
015            IF CRCAT2 EQUALS '0' MOVE 2  TO CAT.
015            IF CRCAT2 EQUALS '1' MOVE 3  TO CAT.
015            ADD SQFT TO SQFTA(PT, 16).
015            GO TO SUM-CAT.
015        NON-VEH.
015            IF CONCO EQUALS 'C' OR 'B' MOVE 11 TO CAT
```

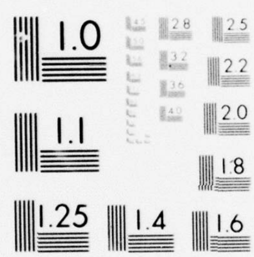
```
015          ELSE MOVE 10 TO CAT.
015          IF CRCAT2 EQUALS '0' MOVE 1  TO CAT.
015          IF CRCAT2 EQUALS '1' MOVE 5  TO CAT.
015          IF CRCAT2 EQUALS '3' MOVE 13 TO CAT.
015          IF CRCAT2 EQUALS '7' MOVE 14 TO CAT.
015          GO TO SUM-CAT.
015      AMMO.
015          MOVE 15 TO CAT.
015      SUM-CAT.
015          ADD STON TO STONA(PT, CAT).
015          ADD SQFT TO SQFTA(PT, CAT).
015          IF CAT IS LESS THAN 14 ADD STON TO STONA(PT, 16)
015          ELSE                   ADD STON TO STONA(PT, 17).
015          GO TO LOAD-F5.
015      FIROLL.
015          IF F5SEQ EQUALS 1 GO TO FISAVE.
015      PRINT-SUM.
015          IF UNCOUNT EQUALS 10 MOVE ZERO TO UNCOUNT
015          WRITE PRINT-LINE FROM HEAD-LINE
015          AFTER ADVANCING TOP-OF-PAGE LINES.
015          ADD 1 TO UNCOUNT.
015          IF PT EQUALS 1
015          MOVE ' TO BE MOVED'      TO NAME     IN LINE2
015          MOVE '1'                 TO LINE-1   IN LINE1
015          MOVE '2'                 TO LINE-2   IN LINE2
015          ELSE
015          MOVE ' PREPO       '     TO NAME     IN LINE2
015          MOVE ZERO                TO STR      IN LINE1
015          MOVE ZERO                TO PAX      IN LINE2
015          MOVE '3'                 TO LINE-1   IN LINE1
015          MOVE '4'                 TO LINE-2   IN LINE2.
015          MOVE STONA(PT, 1)        TO STON-AD  IN LINE1.
015          MOVE STONA(PT, 2)        TO STON-BD  IN LINE1.
015          MOVE STONA(PT, 3)        TO STON-B1  IN LINE1.
015          MOVE STONA(PT, 4)        TO STON-A1  IN LINE1.
015          MOVE STONA(PT, 5)        TO STON-J1  IN LINE1.
015          MOVE STONA(PT, 6)        TO STON-B2C IN LINE1.
015          MOVE STONA(PT, 7)        TO STON-B2D IN LINE1.
015          MOVE STONA(PT, 8)        TO STON-A2C IN LINE1.
015          MOVE STONA(PT, 9)        TO STON-A2D IN LINE1.
015          MOVE STONA(PT, 10)       TO STON-J2D IN LINE1.
015          MOVE STONA(PT, 11)       TO STON-J2C IN LINE1.
015          MOVE STONA(PT, 12)       TO STON-A3C IN LINE1.
015          MOVE STONA(PT, 13)       TO STON-J3C IN LINE1.
015          MOVE STONA(PT, 14)       TO STON-J7C IN LINE1.
015          MOVE STONA(PT, 15)       TO STON-M7C IN LINE1.
015          MOVE STONA(PT, 16)       TO STON-UE  IN LINE1.
015          MOVE STONA(PT, 17)       TO STON-AS  IN LINE1.
015          IF PT EQUALS 1
015          WRITE PRINT-LINE FROM LINE1 AFTER ADVANCING 2 LINES
015          ELSE
015          WRITE PRINT-LINE FROM LINE1 AFTER ADVANCING 1 LINES.
015          MOVE SQFTA(PT, 1)        TO SQFT-AD  IN LINE2.
015          MOVE SQFTA(PT, 2)        TO SQFT-BD  IN LINE2.
015          MOVE SQFTA(PT, 3)        TO SQFT-B1  IN LINE2.
015          MOVE SQFTA(PT, 4)        TO SQFT-A1  IN LINE2.
015          MOVE SQFTA(PT, 6)        TO SQFT-B2C IN LINE2.
```

```
015                MOVE SQFTA(PT, 7)        TO SQFT-B2D IN LINE2.
015                MOVE SQFTA(PT, 8)        TO SQFT-A2C IN LINE2.
015                MOVE SQFTA(PT, 9)        TO SQFT-A2D IN LINE2.
015                MOVE SQFTA(PT, 12)       TO SQFT-A3C IN LINE2.
015                MOVE SQFTA(PT, 16)       TO SQFT-NSDA.
015                MOVE SQFTA(PT, 17)       TO SQFT-VEH IN LINE2.
015                WRITE PRINT-LINE FROM LINE2.
015                ENTER FORTRAN SDFTU SUBROUTINE.
015            FISAVE.
015                MOVE SRC19  IN LOG-REC TO SRC1-9 IN LINE1.
015                MOVE SRC11-12          TO SRCPAR IN LINE1.
015                MOVE LEVCO             TO ULC    IN LINE1.
015                MOVE UTC    IN LOG-REC TO UTC    IN LINE1.
015                MOVE UNSTR             TO STR    IN LINE1.
015                MOVE UIC    IN LOG-REC TO UIC    IN LINE2.
015                MOVE PERTO IN LOG-REC  TO PAX    IN LINE2.
015                MOVE 2  TO PT.
015                PERFORM ZERO-ARRAYS VARYING CAT FROM 1 BY 1
015                UNTIL CAT IS GREATER THAN 18.
015                MOVE 1  TO PT.
015                PERFORM ZERO-ARRAYS VARYING CAT FROM 1 BY 1
015                UNTIL CAT IS GREATER THAN 18.
015                GO TO LOAD-F5.
015            ZERO-ARRAYS.
015                MOVE ZEROS TO STONA(PT, CAT) SQFTA(PT, CAT).
015            LAST-SUM.
015                PERFORM PRINT-SUM.
015                ENTER FORTRAN EOF12 SUBROUTINE.
015            END-LOAD.
015                EXIT.
015            KEY-ERR.
015                MOVE F5KEY TO NUMB.
015                MOVE 'INVALID KEY FOR FIF2 FILE' TO MESSAGE.
015                MOVE ERROR-WORK TO PRINT-LINE.
015            PRINT-OUT.
015                WRITE PRINT-LINE.
015            CLOSE-FILES.
015                IF LOAD5  CLOSE F5 USING F5-INFO.
015                MOVE 'F5 INFORMATION AREA' TO PRINT-LINE.
015                PERFORM PRINT-OUT.
015                MOVE CORRESPONDING F5-INFO TO F5-INFORMATION.
015                MOVE  F5-INFORMATION        TO PRINT-LINE.
015                PERFORM PRINT-OUT.
015                CLOSE PRINT-FILE.
015                STOP RUN.

END ELT.

@HDG            MRG PROGRAM GEO          ••UNCLASSIFIED•• •L,0
```

```
@ELT,L   58CAMP.GEO
ELT007 RLIB70 08/16-07:43:31-(20,)
       016          PARAMETER NG=9000
       016          IMPLICIT INTEGER (A-Z)
       016          REAL ACC
       016          DIMENSION THTR(7),AVAIL(7),DEST(7,2),AVT(7),
       016         *  XTPSN(20),XDSTG(20),XDSTN(20),XAVL(20),XRDD(20),XMODE(20),
       016         *  XTHR(20),XORGG(20),XORGN(20)
       016          DIMENSION SRC(2),UNTDS(2),MDFI(17)
       016          EQUIVALENCE   (MDFI( 1),SRC),        @ 2A6
       016         *              (MDFI( 3),UIC),        @  A6
       016         *              (MDFI( 4),TPSN),       @  I5
       016         *              (MDFI( 5),ROB ),       @  A1
       016         *              (MDFI( 6),GO24),       @  A6
       016         *              (MDFI( 7),GD24),       @  A6
       016         *              (MDFI( 8),ONOD),       @  I3
       016         *              (MDFI( 9),DNOD),       @  I3
       016         *              (MDFI(10),RDD ),       @  I3
       016         *              (MDFI(11),AVL ),       @  I3
       016         *              (MDFI(12),MODE),       @  A1
       016         *              (MDFI(13),THR ),       @  I1
       016         *              (MDFI(14),MCODE),      @  I1
       016         *              (MDFI(15),UNTDS),      @ 2A6
       016         *              (MDFI(17),FPSAV)       @  I2
       016          DIMENSION FASREC(52),F(52),LOCNAM(2),FNAME(3)
       016         *, LIST(3,NG),FIRST(2),LAST(2)
       016          EQUIVALENCE (F,FASREC)
       016          EQUIVALENCE (LN1,LOCNAM(1)),(LN2,LOCNAM(2))
       016    C
       016          DATA FIRST/2*1/
       016          DATA FNAME/5H81FAS
       016         *,         6H81GEO
       018         *,         6H81NODE/
       016          DATA IFL/17/
       016          DIMENSION FILE(2)
       016          DATA FILE /'GEO','NODE'/
       016          DATA T/'@@@@@T'/, B/'@@@@@B'/, E/'@@@@@E'/, R/'@@@@@R'/,
       016         * ONE/'@@@@@1'/, BLANK/'      '/, MZERO/-0/, M/'@@@@@M'/,
       016         * MED/'@@@MED'/, DC/'@@@@DC'/, FOUR/'@@@@@4'/, X8/'@@@@X8'/,
       019         * TWO/'@@@@@2'/, THREE/'@@@@@3'/, NINE/'@@@@@9'/
       016    C
       016          DIMENSION KEY(13)
       016          DATA KEY /1,1,36,0,0,1,          @ SRC(1)
       016         *          2,1,36,0,0,2,          @ SRC(2)
       016         *          99999/
       016          LOGICAL ERROR
       016    C
       016    C  DEFINE CHARACTER FUNCTIONS
       016    C
       016          DEFINE C1(X) = FLD(0,6,X)
       016          DEFINE C2(X) = FLD(6,6,X)
       016          DEFINE C3(X) = FLD(12,6,X)
       016          DEFINE C4(X) = FLD(18,6,X)
       016          DEFINE C5(X) = FLD(24,6,X)
       016          DEFINE C6(X) = FLD(30,6,X)
       016          DEFINE C12(X) = FLD(0,12,X)
       016          DEFINE C13(X) = FLD(0,18,X)
```

AD-A048 554 ARMY CONCEPTS ANALYSIS AGENCY BETHESDA MD F/G 15/7
COMPUTER ASSISTED MATCH PROGRAM (CAMP), (U)
AUG 76 G L MARTIN, E R MONTAGNE
UNCLASSIFIED CAA-D-76-5 NL

3 OF 4
AD A048554



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
016          DEFINE C14(X) = FLD(0,24,X)
016          DEFINE C15(X) = FLD(0,30,X)
016          DEFINE C23(X) = FLD(6,12,X)
016          DEFINE C24(X) = FLD(6,18,X)
016          DEFINE C25(X) = FLD(6,24,X)
016          DEFINE C26(X) = FLD(6,30,X)
016          DEFINE C34(X) = FLD(12,12,X)
016          DEFINE C35(X) = FLD(12,18,X)
016          DEFINE C36(X) = FLD(12,24,X)
016          DEFINE C45(X) = FLD(18,12,X)
016          DEFINE C46(X) = FLD(18,18,X)
016          DEFINE C56(X) = FLD(24,12,X)
016    C
016    C
016    C   CREATES MDFI FILE IN SRC SEQUENCE
016    C
016    C   ENTER NUMBER OF THEATERS
016          READ(5,1000) NTH
016    1000  FORMAT(I1)
016    C   ENTER DESTINATION GEO AND NODE, AVAIL DAY
016          READ(5,1001) (THTR(K),AVAIL(K),DEST(K,1),DEST(K,2),AVT(K),
016        •  K = 1,NTH)
016    1001  FORMAT(I1,1X,I3,1X,A4,1X,I3,1X,I3)
016    C   ENTER TPSN LIST FOR EXCEPTIONS
016          READ(5,1020) NTPSNX
016          IF(NTPSNX .EQ. 0) GO TO 1109
016          READ(5,1002) (XTPSN(I),XDSTG(I),XDSTN(I),XAVL(I),XRDD(I),XMODE(I),
016        •           XTHR(I),XORGG(I),XORGN(I),I=1,NTPSNX)
016    1002  FORMAT((I5,1X,A4,3(1X,I3),1X,A1,1X,I1,1X,A4,1X,I3))
016    1020  FORMAT(I2)
016    C
016    C   PRINT OUT INPUT
016    1109  PRINT 1110, NTH,NTPSNX
016    1110  FORMAT(' NUMBER OF THEATERS    ',I2/
016        •       ' NUMBER OF EXCEPTIONS ',I2)
016          PRINT 1120, (THTR(K),AVAIL(K),DEST(K,1),DEST(K,2),AVT(K),K=1,NTH)
016    1120  FORMAT(/' THTR AVAIL DEST(G) DEST(N) AVL(T)'/
016        •   (3X,I1,3X,I3,3X,A4,4X,I3,3X,I3))
016          IF(NTPSNX .EQ. 0) GO TO 1131
016          PRINT 1130, (XTPSN(I),XDSTG(I),XDSTN(I),XAVL(I),XRDD(I),XMODE(I),
016        •   XTHR(I),XORGG(I),XORGN(I),I=1,NTPSNX)
016    1130  FORMAT(/' TPSN  DEST(G)  DEST(N)  AVL  RDD  MODE  THR',
016        •  '  ORIG(G)  ORIG(N)'/
016        •   (1X,I5,3X,A4,4X,I3,5X,I3,2X,I3,4X,A1,4X,I1,4X,A4,5X,I3))
016    C   ERROR CHECK
016    1131  IF(NTH .GT. 0 .AND. NTH .LE. 7) GO TO 1150
016          PRINT 1140, NTH
016    1140  FORMAT(1X,I2,' THEATERS INPUT. MUST BE BETWEEN 0 - 7')
016          IER = 1
```

```
016  50      IG=IG+1
016          READ(15,1,END=60) (LIST(I,IG),I=1,3)
016          GOTO50
016  C
016  C  ENTER NODE TABLE (LIST)
016  60      LAST(1)=IG
016          IG=IG-1
016          FIRST(2)=IG
016          IR=IG
016          PRINT 95,IR,FNAME(2)
016       95 FORMAT(I8,' RECORDS ON FILE '2A6)
016  C
016  70      IG=IG+1
016          READ(16,1003,END=79) LIST(1,IG),LIST(3,IG),LIST(2,IG)
016  1003    FORMAT(A6,6X,A6,3X,I3)
016          C46(LIST(1,IG)) = 0
016        1 FORMAT(12A6)
016          GOTO70
016  75      PRINT 76, IG
016  76      FORMAT(I6,' ENTRIES IN GEO AND NODE. CAN ONLY HAVE 9000')
016          STOP
016  79      CONTINUE
016          IF(IG .GT. 9001) GO TO 75
016          LAST(2)=IG
016          IR=IG-IR-1
016          PRINT 95,IR,FNAME(3)
016  C
016  C
016          PRINT 97
016       97 FORMAT('         UNITS        ACTIVE       NO FIND       RESERVE',
016         *'      NO FIND    AVE ACC/U')
016  C
016  C  SET TO SORT OUTPUT ON SRC
016          CALL SOPEN3($100,$990,17,2,KEY)
016  C  MAIN LOOP
016  C
016  100     CONTINUE
016          READ(IFL,END=900) FASREC
017  C    THEATER CODE
016          THR = C1(F(8))-48
016          DO 101 ITH = 1,NTH
016          IF(THR .EQ. THTR(ITH)) GO TO 102
016  101     CONTINUE
016          GO TO 100
016  C    SKIP IF UNIT IS DISPLAY ONLY
016  102     IF(C56(F(12)) .NE. DC) GO TO 100
016  C    IF SRC IS BLANK /TDA UNIT/ DO NOT CONSIDER
016          SRC(1) = C26(F(23))*2**6 + C1(F(24))
016          IF(SRC(1) .EQ. BLANK) GO TO 100
016          SRC(2) = C24(F(24))*2**18 + C6(F(24))*2**12 + C1(F(25))*2**6
016         *            + '00000 '
016              COMPO = C3(F(1))
016              UIC = C46(F(1))*2**18 + C13(F(2))
016              IF(COMPO .EQ. FOUR .AND. C12(UIC) .EQ. X8) GO TO 100
016              IF(COMPO .EQ. ONE .AND. C4(F(14)) .EQ. B) GO TO 100
016          KU =KU + 1
016  C  PROGRESS REPORT
```

```
016          IF(KU/500*500.EQ.KU) PRINT 99,KU,KA,KAN,KR,KRN,ACC
016          DECODE(10,1004,F(30)) TPSN
016     1004 FORMAT(5X,I5)
016     C
016     C
016     C
016     C      COMPUTE /MCODE/
016     C         0=IN COUNTRY, 1=PREPO, 2=CONUS ORIENTED ON NATO, 3=CONUS OTHER
016     C
016          ROB = BLANK
016          MCODE = 3
016          IF(COMPO .NE. ONE) GO TO 200
016          P3 = C6(F(22))
016          IF(P3.EQ.T .OR. P3.EQ.E) C1(ROB) = P3
016          IF(P3 .EQ. B) C1(ROB) = E
016          P2 = C5(F(22))
016          IF(P2 .EQ.R) C1(ROB) = P2
016          IF(C46(F(22)) .EQ. MED) C1(ROB) = M
016          IF(ROB .NE. BLANK) MCODE = 1
019     C    CHECK ADCO1 FIELD  - POS 119
020          IF(C5(F(20)) .EQ. E .OR. C5(F(20)) .EQ. NINE)  MCODE = 0
016          IF(MCODE .NE. 1) ROB = BLANK
016     C
016     C
016     C  ACTIVE UNIT
016     C  LOCCO  STNNM
016     84   KA=KA+1
016          LN1 = C46(F(26))
016          C13(LN1) = C46(F(15))
016          IF(C1(LN1).GE.48 .AND. C1(LN1).LE.57) C1(LN1) = BLANK
016          LN2 = F(27)
016          LG = LOCATE(LOCNAM,1)
016          IF(LG.LE.0) KAN=KAN+1
016     C    IF(LG .EQ. 0) PRINT 91, FILE(1),LOCNAM,UIC
016     C
016     C  LOCCO
016          C13(LOCCO) = C13(LN1)
016          LN = LOCATE(LOCCO,2)
016          IF(LN.GT.0) GOTO500
016     C
016     C  DIAGNOSTIC
016          PRINT 91, FILE(2),LOCNAM,LOCCO,UIC
016        91 FORMAT(' ***ACTIVE UNIT AREA NOT FOUND IN ',A5,'FILE ',
016       *              3X,2A6,2(2X,A6))
016     C
016          GOTO500
016     C
016     C
016     C
016     C
016     C  RESERVE UNIT
016     200  CONTINUE
016          KR=KR+1
016     C
016     C  MBLOC  MBSTA
016          LN1 = C6(F(16))*2**30 + C12(F(17))*2**18 + C56(F(17))*2**6
016       *              + C1(F(18))
```

```
016          IF(C1(LN1).GE.48 .AND. C1(LN1).LE.57) C1(LN1) = BLANK
016          LN2 = C26(F(18))*2**6 + C1(F(19))
016          LG = LOCATE(LOCNAM,1)
016    C     IF(LG .EQ. 0) PRINT 92, FILE(1),LOCNAM,UIC
016          IF(LG.LE.0) KRN=KRN+1
016    C
016    C  MBLOC
016          C13(MBLOC) = C13(LN1)
016          LN = LOCATE(MBLOC,2)
016          IF(LN.GT.0) GOTO500
016    C
016    C  LOCCO
016          C13(LOCCO) = C46(F(15))
016          IF(C1(LOCCO).GE.48 .AND. C1(LOCCO).LE.57) C1(LOCCO) = BLANK
016          LN = LOCATE(LOCCO,2)
016          IF(LN.GT.0) GOTO500
016    C
016    C  DIAGNOSTIC
016          PRINT 92, FILE(2),LOCNAM,LOCCO,UIC
016       92 FORMAT(' *** RESERVE UNIT AREA NOT FOUND IN ',A5,'FILE ',
016         *               3X,2A6,2(2X,A6))
016    C
016    C
016    C
016    C
016    C
016    C  BUILD MOVEMENT DATA FILE RECORD
016    500   CONTINUE
016    C
016    C
016    C
016    C  RDD, AVAIL
016          DECODE(6,1005,F(7)) RDD
016    1005  FORMAT(2X,I3)
016          AVL = AVAIL(ITH)
016              IF(TPSN .LT. 20000 .AND. (COMPO.EQ.TWO .OR. COMPO.EQ.THREE))
016         *                    AVL = AVT(ITH)
016    C
016    C  MODE, ULC
016          MODE = 'P'
016          IF(MCODE .EQ. 0) MODE = 'Z'
016          IF(MCODE .EQ. 1) MODE = 'A'
016          ULC = C6(F(34))*2**30 + C12(F(35))*2**24
016    C
016    C  UNTDS (1-12)
016          UNTDS(1) = C36(F(35))*2**12 + C12(F(36))
016          UNTDS(2) = C36(F(36))*2**12 + C12(F(37))
016    C
016    C  ORIGIN: G024, ARLOC (1-2)     CK MATCH RESULTS
016          IF(LN .EQ. 0) GO TO 950
016          IF(LG.EQ.0) LG=LN
016    C
016          G024 = LIST(3,LG)
016          ONOD = LIST(2,LN)
016    C
016    C  DESTINATION
016          GD24 = DEST(ITH,1)
```

```
016            DNOD = DEST(ITH,2)
016      C
016            IF(MCODE .NE. 0) GO TO 1040
016      C   IN COUNTRY - SET ORIGIN TO DESTINATION. NO EXCEPTION LIST
016            GO24 = GD24
016            GO TO 1090
016      C   SEE IF UNIT IS ON EXCEPTION LIST
016      1040  DO 1050 I = 1,NTPSNX
016            IF(TPSN .EQ. XTPSN(I)) GO TO 1060
016      1050  CONTINUE
016            GO TO 1090
016      1060  IF(XDSTG(I) .NE. BLANK) GD24 = XDSTG(I)
016            IF(XDSTN(I) .NE. MZERO) DNOD = XDSTN(I)
016            IF(XRDD (I) .NE. MZERO) RDD  = XRDD (I)
016            IF(XAVL (I) .NE. MZERO) AVL  = XAVL (I)
016            IF(XMODE(I) .NE. BLANK) MODE = XMODE(I)
016            IF(XTHR (I) .NE. MZERO) THR  = XTHR(I)
016            IF(XORGG(I) .NE. BLANK) GO24 = XORGG(I)
016            IF(XORGN(I) .NE. MZERO) ONOD = XORGN(I)
016      C  WRITE RECORD
016      C DEBUG TO UNIT 7
016      1090  WRITE(7,1100) MDF1
016      1100  FORMAT(1X,2A6,1X,A6,1X,I5,1X,A1,2(1X,A6),4(1X,I3),1X,A1,2(1X,I1),
016           *       1X,2A6,1X,A1)
016            CALL SRREL(MDF1,17)
016            ACC = ((KU-1)*ACC+KC)/KU
016            KC=0
016            GOTO100
016      C
016      C
016      900   ENDFILE 21
016            PRINT 99,KU,KA,KAN,KR,KRN,ACC,KF
016         99 FORMAT()
016      C   SORT MDF1 ACCORDING TO SRC
016      980   CALL SSORT
016      990   CALL SRRET(MDF1,17,$995)
016            WRITE(29) MDF1
016            GO TO 990
016      995   ENDFILE 29
016            IF(ERROR) RETURN 0
016            STOP
016      C
016      C
016      C   CAN'T OUTPUT THIS MDF - SAVE FAS RECORD FOR REPROCESSING
016      950   KU = KU - 1
016            KF = KF + 1
016            ERROR = .TRUE.
016            GO TO 100
016      C
016      C
016      C*************************
016            FUNCTION LOCATE(ARR,N)
016            DIMENSION ARR(2)
016      C
016            LOCATE=0
016            HI=LAST(N)
016            LO=FIRST(N)
```

```
016  C
016  C  BINARY SEARCH
016  C
016  100    LP=LO+(HI-LO)/2
016      99 FORMAT()
016         KC=KC+1
016    1    FORMAT(2X,2A6,2X,2A6,I6)
016         IF(ARR(1)-LIST(1,LP)) 200,110,400
016  110    IF(N.EQ.2) GOTO300
016         IF(ARR(2)-LIST(2,LP)) 200,300,400
016  C
016  C  HIGH
016  200    IF(LP-LO.LE.1) RETURN
016         HI=LP
016         GOTO100
016  C
016  C  HIT
016  300    LOCATE=LP
016         RETURN
016  C
016  C  LOW
016  400    IF(HI-LP.LE.1) RETURN
016         LO=LP
016         GOTO100
016         END
```

END ELT.

@HDG MRG PROGRAM LOG **UNCLASSIFIED** .L,0

```
@ELT,L   58CAMP.LOG
ELT007 RLIB70 08/16-07:43:36-(0,)
000          IMPLICIT INTEGER (A-X)
000          DIMENSION SRC(2),UNTDS(2),MDFI(17),MDFIA(20)
000          EQUIVALENCE (MDFI,MDFIA)
000          EQUIVALENCE  (MDFI( 1),SRC),      @ 2A6
000     *                 (MDFI( 3),UIC),      @  A6
000     *                 (MDFI( 4),TPSN),     @  I5
000     *                 (MDFI( 5),ROB ),     @  A1
000     *                 (MDFI( 6),GO24),     @  A6
000     *                 (MDFI( 7),GD24),     @  A6
000     *                 (MDFI( 8),ONOD),     @  I3
000     *                 (MDFI( 9),DNOD),     @  I3
000     *                 (MDFI(10),RDD ),     @  I3
000     *                 (MDFI(11),AVL ),     @  I3
000     *                 (MDFI(12),MODE),     @  A1
000     *                 (MDFI(13),THR ),     @  I1
000     *                 (MDFI(14),MCODE),    @  I1
000     *                 (MDFI(15),UNTDS),    @ 2A6
000     *                 (MDFI(17),FPSAV)     @  I2
000          DIMENSION P(22,4)
000          EQUIVALENCE (P,TUCHA), (P(1,3),POMCUS)
000          DIMENSION TUCHA(22,2),TSRC(2),TNAM(2)
000          EQUIVALENCE (TUCHA( 1,1),   TSRC),   (TUCHA( 1,2),  PUIC),
000     *                (TUCHA( 3,1),    ULC),   (TUCHA( 3,2),  TNAM),
000     *                (TUCHA( 4,1),    UTC),
000     *                                         (TUCHA( 5,2),   PAX),
000     *                (TUCHA( 6,1),ZVEHNA),    (TUCHA( 6,2),VEHNAT),
000     *                (TUCHA( 7,1),ZNSDNA),    (TUCHA( 7,2),NSDNAT),
000     *                (TUCHA( 8,1),ZNSDOU),    (TUCHA( 8,2),NSDOUT),
000     *                (TUCHA( 9,1),ZVEHOU),    (TUCHA( 9,2),VEHOUT),
000     *                (TUCHA(10,1),ZNVOUT),
000     *                (TUCHA(11,1),ZNSOVC),    (TUCHA(11,2),NSDOVC),
000     *                (TUCHA(12,1),ZNSOVN),    (TUCHA(12,2),NSDOVN),
000     *                (TUCHA(13,1),ZVHOVC),    (TUCHA(13,2),VEHOVC),
000     *                (TUCHA(14,1),ZVHOVN),    (TUCHA(14,2),VEHOVN),
000     *                (TUCHA(15,1),ZNVOVN),
000     *                (TUCHA(16,1),ZNVOVC),
000     *                (TUCHA(17,1),ZVEHBC),    (TUCHA(17,2), VEHBC),
000     *                (TUCHA(18,1), ZNVBC)
000          DIMENSION POMCUS(22,2)
000          EQUIVALENCE                           (POMCUS( 5,2),  PAXP),
000     *                (POMCUS( 6,1),YVEHNA),    (POMCUS( 6,2),VEHNAP),
000     *                (POMCUS( 7,1),YNSDNA),    (POMCUS( 7,2),NSDNAP),
000     *                (POMCUS( 8,1),YNSDOU),    (POMCUS( 8,2),NSDOUP),
000     *                (POMCUS( 9,1),YVEHOU),    (POMCUS( 9,2),VEHOUP),
000     *                (POMCUS(10,1),YNVOUT),
000     *                (POMCUS(11,1),YNSOVC),    (POMCUS(11,2),NSOVCP),
000     *                (POMCUS(12,1),YNSOVN),    (POMCUS(12,2),NSOVNP),
000     *                (POMCUS(13,1),YVHOVC),    (POMCUS(13,2),VHOVCP),
000     *                (POMCUS(14,1),YVHOVN),    (POMCUS(14,2),VHOVNP),
000     *                (POMCUS(15,1),YNVOVN),
000     *                (POMCUS(16,1),YNVOVC),
000     *                (POMCUS(17,1),YVEHBC),    (POMCUS(17,2),VEHBCP),
000     *                (POMCUS(18,1), YNVBC)
000          DIMENSION PORTS(2),PLANID(7),MORSW(6),MORSWP(6)
000          EQUIVALENCE (MORSW(1),ASUPLY),     (MORSWP(1),PASUPL),
```

```
000        •              (MORSW(2),   BULK),      (MORSWP(2), PBULK),
000        •              (MORSW(3),   OVER),      (MORSWP(3), POVER),
000        •              (MORSW(4),    OUT),      (MORSWP(4);  POUT),
000        •              (MORSW(5),    NAT),      (MORSWP(5),  PNAT),
000        •              (MORSW(6),ACAMMO),       (MORSWP(6),PACAMM)
000          DIMENSION SEQ(7),THTR(7),TSAV(2),ZBASR(2),ZAASR(2),THTRID(7),
000        •   ICT(7)
000          LOGICAL ENDT, ENDP, ENDM, ERROR, FIND
000          DIMENSION SSRC(4,99)
000          DIMENSION IZERO(44)
000          DATA IZERO /44*0/
000          DIMENSION  KEY(7), KEY1(13)
000          DATA KEY /3,1,36,0,0,1,  @ UIC
000        •          99999/
000          DATA KEY1 /1,1,36,0,0,1,    @ SRC(1)
000        •            2,1,36,0,0,2,    @ SRC(2)
000        •            99999/
000          DATA MD/'Z'/, PAVL/999/, PROD/0/
000    C
000          DEFINE C3(X) = FLD(12,6,X)
000    C
000          READ(5,1) IN,NTH,THTR,THTRID,PLANID
000    1     FORMAT(2(I1,1X),2(7I1,1X),7A1)
000          READ(5,2) PORTS,ZBASR,ZAASR        @ DEBARKATON AND ASR(LBS)
000    2     FORMAT(2(A4,1X),4(F6.0,1X))
000          READ(5,3) NSRCX
000    3     FORMAT(I2)
000          IF(NSRCX .EQ. 0) GO TO 44
000          READ(5,4) ((SSRC(I,J),I=1,4),J=1,NSRCX)
000    4     FORMAT(4A6)
000    C   SORT SRC INPUT - MATCH UPMDF1 INPUT
000          CALL SOPEN1($40,$42,4,2,KEY1)
000    40    DO 41 I = 1,NSRCX
000    41    CALL SRREL(SSRC(1,I),4)
000          CALL SSORT
000    42    I = 1
000    43    CALL SRRET(SSRC(1,I),4,$44)
000          I = I + 1
000          GO TO 43
000    44    CONTINUE
000    C
000          IF(IN .LE. 0 .OR. IN .GT. 7) IER = 1
000          IF(NTH .LE. 0 .OR. NTH .GT. 7) IER = IER + 2
000          DO 5 I = 1,IN
000          IF(THTRID(I) .EQ. 0) IER = IER + 4
000          IF(I .GT. NTH) GO TO 5
000          IF(PLANID(I) .EQ. ' ') IER = IER + 8
000    5     CONTINUE
000          PRINT 6, IN,NTH
000    6     FORMAT( ' NUMBER OF THEATERS ',I2/
000        •        ' NUMBER OF SETS      ',I2)
000          PRINT 7
000    7     FORMAT(' THTR  SET  PLANID')
000          DO 8 I = 1,IN
000          J = THTRID(I)
000    8     PRINT 9, THTR(I),THTRID(I),PLANID(J)
000    9     FORMAT(3X,I1,4X,I1,6X,A1)
```

```
000          PRINT 10
000    10    FORMAT(6X,'AIRPORT',6X,'SEAPORT',10X,'NON-POMCUS ASR',15X,
000        •    'POMCUS ASR'/33X,'SUPPLY',9X,'AMMO',7X,'SUPPLY',9X,'AMMO')
000          PRINT 11, PORTS,(ZBASR(I),ZAASR(I),I=1,2)
000    11    FORMAT(8X,A4,9X,A4,5X,2(F9.3,5X,F9.3,3X))
000          IF(NSRCX .EQ. 0) GO TO 16
000          PRINT 17, ((SSRC(I,J),I=1,4),J=1,NSRCX)
000    17    FORMAT(/' SRC LIST'/(2X,2A6,2X,2A6))
000    16    IF(IER .EQ. 0) GO TO 98
000          IF(MOD(IER,2) .NE. 0) PRINT 12
000    12    FORMAT(' *****ERROR***** NUMBER OF THEATERS')
000          IER = IER/2
000          IF(MOD(IER,2) .NE. 0) PRINT 13
000    13    FORMAT(' *****ERROR***** NUMBER OF SETS')
000          IER = IER/2
000          IF(MOD(IER,2) .NE. 0) PRINT 14
000    14    FORMAT(' *****ERROR***** SET NUMBERS')
000          IER = IER/2
000          IF(MOD(IER,2) .NE. 0) PRINT 15
000    15    FORMAT(' *****ERROR***** PLANID ERROR')
000          RETURN 0
000    98    CONTINUE
000    C
000    C   READ MDF1. FIND MATCH ON TUCHA (AND POMCUS IF MCODE = 1). FILES IN SRC SEW
000    C    PRODUCE UNIT MORSA CARDS AND MDF2
000    C   29 = MDF1, 28 = TUCHA, 27 = POMCUS, 26 = MDF2, 10 = MORSA CARDS
000    C
000          CALL SOPEN2($99,$430,20,3,KEY)
000    99    POM = 1
000          FIND = .TRUE.
000          DO 999 I = 1,2
000    C   CONVERT TO TONS
000          ZBASR(I) = ZBASR(I)/2000.
000    999   ZAASR(I) = ZAASR(I)/2000.
000    C
000          READ(28,151) TUCHA
000    100   READ(29,END=400) MDF1
000          IF(ENDT) GO TO 180
000          NU = NU + 1
000    110   IF(SRC(1) - TSRC(1)) 180,120,150
000    C   CHECK ONLY 7,8,9 OF SRC(2)
000    120   IF(FLD(0,18,SRC(2)) - FLD(0,18,TSRC(2))) 180,200,150
000    150   READ(28,151,END=1030) TUCHA
000    151   FORMAT(A6,A5,3X,A3,2X,A5,18F6.1/A6,A5,1X,2A6,18I6)
000          GO TO 110
000    C   NO MATCH FOR MDF1 UNIT ON TUCHA
000    180   IF(SRC(1) .NE. TSAV(1)) GO TO 185
000          IF(SRC(2) .NE. TSAV(2)) GO TO 185
000          GO TO 190
000    185   PRINT 181, SRC,UIC,TSRC ,NU
000    181   FORMAT( ' NO MATCH FOR SRC ',2A6,'UIC ',A6,' ON TUCHA FILE ',2A6,
000        •  I6)
000          TSAV(1) = SRC(1)
000          TSAV(2) = SRC(2)
000    190   ERROR = .TRUE.
000          NCT = NCT + 1
000          GO TO 100
```

```
000     C
000     C     MATCHED SRC'S. IS THIS A POMCUS UNIT?
000     200     IF(MCODE .EQ. 1) GO TO 300
000     C
000     C     GET THEATER POSITION
000     215     DO 216 ITH = 1,IN
000             IF(THR .EQ. THTR(ITH)) GO TO 217
000     216     CONTINUE
000             PRINT 2000, THTR,SRC,UIC,TPSN,THR
000     2000    FORMAT(7I6/' NO THEATER MATCH FOR '2A6,1X,A6,1X,I5,1X,I3)
000             IF(ENDM) GO TO 430
000             GO TO 100
000     C
000     C     FIX UP SRC IF IT WAS CHANGED IN UPMDF1
000     217     IF(FPSAV .EQ. 0) GO TO 218
000             IF(SRC(1) .NE. SSRC(3,FPSAV)) GO TO 1050
000             IF(SRC(2) .NE. SSRC(4,FPSAV)) GO TO 1050
000             SRC(1) = SSRC(1,FPSAV)
000             SRC(2) = SSRC(2,FPSAV)
000     218     RD = 6
000             ITH = THTRID(ITH)
000             ICT(ITH) = ICT(ITH) + 1
000     C     IF TPSN < 20000 USE SECOND CHARACTER
000             IF(TPSN .LT. 20000) RD = MOD(TPSN/1000,10)
000             IF(TPSN .GE. 30000) RD = 7
000             TAVL = AVL
000             TRDD = RDD
000     C     SET FOR IN-COUNTRY
000             IF(MCODE .EQ. 0) TAVL = 999
000             IF(MCODE .EQ. 0) TRDD = 0
000     C
000             IF(.NOT. FIND) GO TO 209
000     C     SET W CARD DATA
000             TPAX = PAX
000             BULK = ZVEHBC + ZNVBC + .5
000             OVER = ZVHOVC + ZVHOVN + ZNVOVN + ZNVOVC + ZNSOVC + ZNSOVN + .5
000             OUT = ZVEHOU + ZNVOUT + ZNSDOU + .5
000             NAT = ZVEHNA + ZNSDNA + .5
000             FRN = THR*1000 + ONOD
000             NSDAST = ZNSDNA + ZNSDOU + ZNSOVC + ZNSOVN + .5
000             NSDASQ = NSDNAT + NSDOUT + NSDOVC + NSDOVN
000     209     ASUPLY = FLOAT(TPAX) * ZBASR(POM) + .5
000             ACAMMO = FLOAT(TPAX) * ZAASR(POM) + .5
000     C
000     C     WRITE A CARD
000             SEQ(ITH) = SEQ(ITH) + 1
000             WRITE(10,210) SEQ(ITH),PLANID(ITH),RD,UNTDS,ULC,MODE,TAVL,TRDD,
000          *    GD24,GD24,GD24,TPAX,TPSN
000     210     FORMAT(J4,'A',A1,'A      U',I1,2A6,A3,A1,2X,2J3,6X,3A4,2J6,13X)
000     C     WRITE W CARD
000             WRITE(10,220) SEQ(ITH),PLANID(ITH),MORSW,SRC,FRN,UTC,UIC
000     220     FORMAT(J4,'A',A1,'W',6J6,A6,A5,J4,A5,A6,11X)
000             IF(NSDAST + NSDASQ .EQ. 0) GO TO 250
000     C     WRITE F CARD
000             WRITE(10,230) SEQ(ITH),PLANID(ITH),NSDAST,NSDASQ
000     230     FORMAT(J4,'A',A1,'FN',12X,2J6,48X)
000     C
```

```
000  C     WRITE OUT DATA FOR LAYIN
000  250     TOTAL = 0
000          DO 240 I = 1,6
000  240     TOTAL = TOTAL + MORSW(I)
000          WRITE(11,241) TAVL,UIC,RD,SEQ(ITH),GO24,GD24,MODE,TRDD,DNOD,ONOD,
000         •   BULK,OVER,OUT,NAT,NSDASQ,TOTAL,ASUPLY,IZERO(1),TPAX,ACAMMO
000  241     FORMAT(3X,J3,A6,5X,I1,5X,J6,2A4,A1,4X,1X,3J3,2X,5X,10J5)
000  C
000  C     PROCESSING POMCUS?
000          IF(ENDM) GO TO 475
000  C     WRITE MDF2
000          WRITE(26) MDF1,TUCHA,ASUPLY,ACAMMO
000          GO TO 100
000  C
000  C     SAVE POMCUS UNITS
000  300     MDF1A(18) = ULC
000          MDF1A(19) = UTC
000          MDF1A(20) = PAX
000          NP = NP + 1
000          CALL SRREL(MDF1A,20)
000          GO TO 100
000  C
000  C     END OF MDF1
000  400     ENDM = .TRUE.
000          PRINT 1011, ICT,SEQ,NCT
000          POM = 2
000          READ(27,151) P
000  C     SORT POMCUS UNITS ON UIC
000          CALL SSORT
000  430     CALL SRRET(MDF1A,20,$1000)
000          IF(ENDP) GO TO 460
000          FIND = .TRUE.
000  C     FIND MATCH ON POMCUS FILE
000  440     IF(UIC - PUIC) 460,470,450
000  450     READ(27,151,END=1040) P
000          GO TO 440
000  C     NO MATCH
000  460     PRINT 461, SRC,UIC,PUIC
000  461     FORMAT(' NO MATCH FOR SRC ',2A6,' UIC ',A6,' ON POMCUS FILE ',A6)
000          NCT = NCT + 1
000  C     WRITE ZERO RECORD, USE PAX,UTC,ULC FROM TUCHA
000          FIND = .FALSE.
000          TPAX = MDF1A(20)
000          DO 462 I = 1,6
000          MORSW(I) = 0
000  462     MORSWP(I) = 0
000          NSDAST = 0
000          NSDASQ = 0
000          PNSDST = 0
000          PNSDSQ = 0
000          IZERO(3) = ULC
000          IZERO(4) = UTC
000          IZERO(27) = TPAX
000          GO TO 470
000  C
000  C     PROCESS POMCUS UNIT - CREATE 2 SETS OF MORSA CARDS• SMOBSMOD
000  C      DATA ONLY FOR STUFF TO BE MOVED
```

```
000   C
000   470    ULC = MDF1A(18)
000          UTC = MDF1A(19)
000   C    WRITE CARDS FOR STUFF TO BE MOVED
000          GO TO 215
000   C
000   475    WRITE(10,476) SEQ(ITH),PLANID(ITH),ROB
000   C
000          IF(.NOT. FIND) GO TO 477
000   C    COMPUTE AMOUNT ALREADY THERE
000          PBULK = YVEHBC + YNVBC + .5
000          POVER = YVHOVC + YVHOVN + YNVOVN + YNVOVC + YNSOVC + YNSOVN + .5
000          POUT = YVEHOU + YNVOUT + YNSDOU + .5
000          PNAT = YVEHNA + YNSDNA + .5
000          PASUPL = FLOAT(PAXP) * ZBASR(POM) + .5
000          PACAMM = FLOAT(PAXP) * ZAASR(POM) + .5
000          PNSDST = YNSDNA + YNSDOU + YNSOVC + YNSOVN + .5
000          PNSDSQ = NSDNAP + NSDOUP + NSOVCP + NSOVNP
000   C    WRITE CARDS FOR STUFF ALREADY THERE
000   477    SEQ(ITH) = SEQ(ITH) + 1
000          WRITE(10,210) SEQ(ITH),PLANID(ITH),RD,UNTDS,ULC,MD,PAVL,PRDD,GD24,
000         *  GD24,GD24,PAXP,TPSN
000          WRITE(10,220) SEQ(ITH),PLANID(ITH),MORSWP,SRC,FRN,UTC,UIC
000          IF(PNSDST + PNSDSQ .NE. 0) WRITE(10,230) SEQ(ITH),PLANID(ITH),
000         * PNSDST,PNSDSQ
000          WRITE(10,476) SEQ(ITH),PLANID(ITH),ROB
000   476    FORMAT(J4,'A',A1,'FA',1X,A1,70X)
000   C
000          IF(.NOT. FIND) GO TO 480
000   C    MDF2 FILE SHOULD CONTAIN  AMOUNTS TO BE MOVED
000   C
000          WRITE(26) MDF1,TUCHA,ASUPLY,ACAMMO
000          GO TO 430
000   C
000   C    WRITE ZERO RECORD
000   480    WRITE(26) MDF1,IZERO,ASUPLY,ACAMMO
000          GO TO 430
000   C    END OF MDF1 AND POMCUS UNITS
000   C    WRITE OUT SEQ TO BE USED IN NUR FOR REST OF MORSA CARDS
000   1000   DO 1010 I = 1,7
000   1010   MDF1(I) = SEQ(I)
000   C    SET RDD SO IT WILL SORT OUT FIRST IN NUR
000          MDF1(10) = -99999
000          WRITE(26) MDF1,TUCHA,ASUPLY,ACAMMO
000          ENDFILE 26
000          ENDFILE 10
000          ENDFILE 11
000          PRINT 1011, ICT,SEQ,NCT
000   1011   FORMAT('  UNITS IN THEATER SETS ',7I7/
000         *        ' FINAL SEQUENCE NUMBERS ',7I7/
000         *        ' NO MATCHES             ',I7)
000          IF(ERROR) RETURN 0
000          STOP
000   C
000   C    END OF TUCHA FILE. STILL HAVE MDF1 RECORDS LEFT
000   1030   PRINT 1031
000   1031   FORMAT(' END OF TUCHA FILE ')
```

```
000         ENDT = .TRUE.
000         GO TO 180
000   C
000   C    END OF POMCUS FILE. STILL HAVE MDF1A RECORDS LEFT
000   1040  PRINT 1041
000   1041  FORMAT(' END OF POMCUS FILE ')
000         ENDP = .TRUE.
000         GO TO 460
000   C
000   C    BAD MATCH WITH SRC UPDATES
000   1050  PRINT 1051, (MDF1(I),I=1,4),FPSAV,(SSRC(I,FPSAV),I=1,4)
000   1051  FORMAT(' UNIT ',2A6,1X,A6,1X,I5,' NO MATCH TO REVERT BACK TO '
000        *    'ORIGINAL SRC. ENTRY NUMBER IS ',I2/
000        *    'OLD SRC IS ',2A6,' NEW SRC IS ',2A6)
000         RETURN 0
000         END
```

END ELT.

@HDG MRG PROGRAM UPMDF1 ••UNCLASSIFIED•• .L,0

```
@ELT.L    58CAMP.UPMDF1
ELT007 RLIB70 08/16-07:43:39-(0,)
000    C   THIS ROUTINE UPDATES SRC'S BASED ON AN INPUT TABLE - USED IF ERRORS
000    C      IN MATCHING UNITS IN LOG
000           IMPLICIT INTEGER(A-Z)
000           DIMENSION MDF1(17),SRC(4,100),KEY(13),KEY1(13)
000           DATA KEY /1,1,36,0,0,1,
000          •          2,1,36,0,0,2,
000          •          99999/
000           DATA KEY1/1,1,36,0,0,1,        @ SRC(1)
000          •          2,1,36,0,0,2,        @ SRC(2)
000          •          99999/
000           LOGICAL END
000           READ(5,1) NSRC     @ # OF UPDATE ENTRIES
000     1     FORMAT(I2)
000           READ(5,2) ((SRC(J,I),J=1,4),I=1,NSRC) @ OLD AND NEW
000     2     FORMAT(4A6)
000           END = .FALSE.
000    C  SRT INPUT SRC'S
000    C    MDF1 SHOULD BE IN SRC SORT (OUTPUT FROM GEO)
000           CALL SOPEN1($100,$200,4,2,KEY)
000    100    DO 110 I = 1,NSRC
000    110    CALL SRREL(SRC(1,I),4)
000           CALL SSORT
000    200    I = 1
000    210    CALL SRRET(SRC(1,I),4,$250)
000           I = I + 1
000           GO TO 210
000    250    I = 1
000    C    FINAL MDF1 MUST BE IN SRC SORT
000           CALL SOPEN3($300,$410,17,2,KEY1)
000    300    READ(29,END=400) MDF1
000           IF(END) GO TO 330
000    305    IF(MDF1(1) - SRC(1,I)) 330,310,340
000    310    IF(MDF1(2) - SRC(2,I)) 330,320,340
000    320    MDF1(1) = SRC(3,I)
000           MDF1(2) = SRC(4,I)
000           MDF1(17) = I
000    330    CALL SRREL(MDF1,17)
000           GO TO 300
000    340    I = I + 1
000           IF(I .GT. NSRC) GO TO 350
000           GO TO 305
000    350    END = .TRUE.
000           GO TO 330
000    400    CALL SSORT
000    410    CALL SRRET(MDF1,17,$500)
000           WRITE(25) MDF1
000           GO TO 410
000    500    ENDFILE 25
000            STOP
000            END

END ELT.
```

@HDG MRG PROGRAM NUR ••UNCLASSIFIED•• .L,0

```
@ELT,L    58CAMP.NUR
ELT007 RLIB70 08/16-07:43:42-(10,)
006          IMPLICIT INTEGER (A-W)
006          DIMENSION MDF2(63)
006          DIMENSION SRC(2),MDF1(17)
006          EQUIVALENCE  (MDF1( 1),SRC),        @ 2A6
006       •               (MDF1( 3),UIC),        @  A6
006       •               (MDF1( 4),TPSN),       @  I5
006       •               (MDF1(10),RDD ),       @  I3
006       •               (MDF1(11),AVL ),       @  I3
006       •               (MDF1(13),THR ),       @  I1
006       •               (MDF1(14),MCODE)       @  I1
006          DIMENSION TUCHA(22,2)
006          EQUIVALENCE                          (TUCHA( 5,2),    PAX)
006          EQUIVALENCE (MDF2,MDF1), (MDF2(18),TUCHA),
006       •              (MDF2(62),ASUPLY), (MDF2(63),ACAMMO)
006          DIMENSION CDAY(7),BLDUP(7),PRD(8,7),XLEVEL(3,7),LDAY(7),FDAY(7)
006       •,           XSAFE(3,7),GORG(20),NORG(20),XPCT(2,20),XCNSM(3,7,8)
006       •,           XASR(3,7,2),XRPLC(8,7),XSLOPE(3,7),PLANID(7),THTRID(7)
006       •,           XSTREN(7),SEQ(7),XRPLCE(7),XRS(3,7),DDAY(7),PER(7)
006       •,           RT(3),RD(4),RN(2,8),LVL(3),RDAY(7),XRLEV(3,7),AVDAY(7)
006       •,           DEST(2,7),KU(7),THTR(7),THEND(7),XTON(210)
006          EQUIVALENCE (XTON,XCNSM), (XTON(169),XASR)
006          DIMENSION SMOB(8), KEY(7)
006          DATA RT/1HS,1HS,1HP/
006       •,     RD/1HG,1HA,1H2,1HF/
006       •,     RN/'RESUPPLY DRY'
006       •,        'RESUPPLY AMM'
006       •,        'RESUPPLY POL'
006       •,        'REPLACEMENTS'
006       •,        'FILLER      '
006       •,        'PWRS DRY    '
006       •,        'PWRS AMMO   '
006       •,        'PWRS POL    '/
006          DATA LVL/3HPKG,3HPKG,3HPOL/
006          DATA PER/7*1/, BLANK/' '/, MZERO/-0/, IZERO/0/
006          DATA KEY /10,1,36,1,0,1,99999/  @ RDD SORT
006          DIMENSION ADAY(7),XFR(7),BDAY(3,7),XBFR(3,7),MD(4),MOE(7)
006          DATA MD/'P','S','A','Z'/
006          LOGICAL ENDM
006    C
006    C   COMPUTE NON-UNIT MOVEMENT REQUIREMENTS
006    C
006    C
006    C  INITIALIZE
006    C   THEATER DEFINITION
006          READ(5,1001) IN,NTH,THTR,THTRID,PLANID
006    1001  FORMAT(2(I1,1X),2(7I1,1X),7A1)
006    C   DAYS,PERIODS,LEVELS - BY SET
006          READ(5,1002) (K,DDAY(K),RDAY(K),BLDUP(K),CDAY(K),ADAY(K),
006       •       (PRD(L,K),L=1,8),(XLEVEL(L,K),L=1,3),(XSAFE(L,K),L=1,3),
006       •       I = 1,NTH)
006    1002  FORMAT((I1,1X,13(I3,1X),6(F3.0,1X)))
006    C   BASE, FILLER, DEST - BY SET
006          READ(5,1003) (K,(XRS(L,K),L=1,3),XFR(K),(BDAY(L,K),L=1,3),
006       •       (XBFR(L,K),L=1,3),(DEST(L,K),L=1,2),AVDAY(K),I = 1,NTH)
006    1003  FORMAT((I1,1X,3(F9.0,1X),F4.0,1X,3(I3,1X),3(F4.0,1X),A4,2(1X,I3)))
```

```
006   C     REPL (CAS)  BY SET
006           READ(5,1004) (K,(XRPLC(L,K),L=1,8),I=1,NTH)
006   1004  FORMAT((I1,1X,8(F5.0,1X)))
006   C     ASR BY SET
006           READ(5,1005) (K,((XASR(J,K,L),J=1,3),L=1,2),I=1,NTH)
006   1005  FORMAT((I1,1X,6(F6.0,1X)))
006   C     CONSUMPTION - BY TYPE/SET
006           READ(5,1006) ((K,II,(XCNSM(J,K,L),L=1,8),J=1,3),I=1,NTH)
006   1006  FORMAT((2I1,1X,8(F6.0,1X)))
006   C     ORIGINS
006           READ(5,1007) (GORG(L),NORG(L),(XPCT(J,L),J=1,2),L=1,20)
006   1007  FORMAT((A4,1X,I3,1X,2(F4.0,1X)))
006   C
006           PRINT 1100, IN,NTH
006   1100  FORMAT(' NUMBER OF THEATERS ',I2/
006         *         ' NUMBER OF SETS      ',I2)
006           PRINT 1101, (THTR(I),THTRID(I),I=1,IN)
006   1101  FORMAT(/' THTR SET'/(3X,I1,3X,I1/))
006           PRINT 1102
006   1102  FORMAT(' SET PLAN DDAY RDAY BLDUP CDAY ADAY  1   2   3   4   5,
006         * '   6   7   8     BLDUP LEVELS       SAFE LEVELS')
006           DO 1103 K = 1,NTH
006   1103  PRINT 1104, K,PLANID(K),DDAY(K),RDAY(K),BLDUP(K),CDAY(K),
006         * ADAY(K),(PRD(L,K),L=1,8),(XLEVEL(L,K),L=1,3),(XSAFE(L,K),
006         * L=1,3)
006   1104  FORMAT(2X,I1,3X,A1,4X,I3,2X,I3,3X,I3,2(2X,I3),8(1X,I3),
006         * 2(1X,3(1X,F4.0)))
006           PRINT 1105
006   1105  FORMAT(/' SET',14X,'THEATER BASE',14X,'FILLER',5X,'FILLER',10X,
006         * 'FILLER',25X,'DAY'/9X,'DRY',9X,'AMMO',10X,'POL',7X,'PCT',7X,
006         * 'DAYS',12X,'PCTS',9X,'DEST(G)  DEST(N)  AVL')
006           DO 1106 K = 1,NTH
006   1106  PRINT 1107, K,(XRS(L,K),L=1,3),XFR(K),(BDAY(L,K),L=1,3),
006         * (XBFR(L,K),L=1,3),(DEST(L,K),L=1,2),AVDAY(K)
006   1107  FORMAT(2X,I1,1X,3(1X,F12.3),2X,F5.3,2X,3(I3,1X),3(1X,F5.3),
006         * 2X,A4,2(5X,I3))
006           PRINT 1108
006   1108  FORMAT(/' REPLACEMENTS BY TIME PERIOD (MEN/1000/DAY)')
006           DO 1109 K = 1,NTH
006   1109  PRINT 1110, K,(XRPLC(L,K),L=1,8)
006   1110  FORMAT(1X,I1,1X,8(2X,F12.5))
006           PRINT 1111
006   1111  FORMAT(/' ACCOMPANYING SUPPLY RATES (LBS/MAN)'/
006         * 5X,' NON-POMCUS',33X,'POMCUS')
006           DO 1112 K = 1,NTH
006   1112  PRINT 1110, K,((XASR(J,K,L),J=1,3),L=1,2)
006           PRINT 1113
006   1113  FORMAT(/' CONSUMPTION RATES (LBS/MAN/DAY) - BY TIME PERIOD')
006           DO 1114 K = 1,NTH
006   1114  PRINT 1115, K,((XCNSM(J,K,L),L=1,8),J=1,3)
006   1115  FORMAT(2X,I1,2X,'DRY ',8(2X,F12.5)/5X,'AMMO',8(2X,F12.5)/
006         * 5X,'POL ',8(2X,F12.5))
006           PRINT 1116, (GORG(L),NORG(L),(XPCT(J,L),J=1,2),L=1,20)
006   1116  FORMAT(/' ORIG(G)  ORIG(N)  DRY PCT  AMMO PCT'/
006         *  (1X,A4,6X,I3,6X,F5.3,5X,F5.3))
006           DO 1117 I = 1,20
006           XTOT1 = XTOT1 + XPCT(1,I)
```

```
006  1117   XTOT2 = XTOT2 + XPCT(2,I)
006            IER = 0
006            TOT = XTOT1•100• + •5
006            IF(TOT •EQ• 100) GO TO 1119
006            IER = 1
006            PRINT 1118, XTOT1
006  1118      FORMAT('•••••ERROR•••• DRY PERCENTS ADD UP TO ',F5•3)
006  1119      TOT = XTOT2•100• + •5
006            IF(TOT •EQ• 100) GO TO 1121
006            IER = 1
006            PRINT 1120, XTOT2
006  1120      FORMAT('•••••ERROR•••• AMMO PERCENTS ADD UP TO ',F5•3)
007  1121      IF(IN •LE• 7) GO TO 1123
007            PRINT 1122, IN
007            IER = 1
007  1123      IF(NTH •LE• 7) GO TO 1128
007            PRINT 1127, NTH
007            IER = 1
007  1128      DO 1130 I = 1•IN
007  1122      FORMAT('•••••ERROR••••ILLEGAL VALUE FOR NUMBER OF THEATERS ',I2)
007  1127      FORMAT('•••••ERROR••••ILLEGAL VALUE FOR NUMBER OF THEATER SETS',I2)
008            IF(THTRID(I) •GT•0 •AND• THTRID(I) •LE• NTH) GO TO 1125
006            PRINT 1124, THTRID(I)
006  1124      FORMAT('•••••ERROR••••ILLEGAL VALUE FOR SET ',I2)
006  1125      IF(I •GT• NTH) GO TO 1130
006            IF(PLANID(I) •NE• ' ') GO TO 1130
006            IER = 1
006            PRINT 1126, I
006  1126      FORMAT('•••••ERROR••••BLANK PLANID FOR  SET ',I2)
006  1130      CONTINUE
006            IF(IER •NE• 0) RETURN 0
006  C      CONVERT CONSUMPTION AND ASR DATA TO TONS
006            DO 7 I = 1,210
006  7         XTON(I) = XTON(I)/2000•
006            DO 8 K = 1,NTH
006  C      SET FIRST MODE
006            MOE(K) = MD(I)
006  C      FIX CDAY TO GET ALL UNITS ON THAT DAY
006  8         CDAY(K) = CDAY(K) + 1
006  C      SET LAST DAY OF THEATER
006            DO 9 K = 1,NTH
006            DO 92 J = 1,8
006            IF(PRD(J,K) •EQ• MZERO) GO TO 91
006  92        CONTINUE
006            LDAY(K) = PRD(8,K)
006            GO TO 9
006  91        LDAY(K) = PRD(J-1,K)
006  9         CONTINUE
006  C
006  C      WRITE HEADING
006            PRINT 10
006  10        FORMAT(1H1,' DAY  THR  UNITS    STRENGTH  REPLACEMNTS  TYPE     ',
006           • 'DAYS OH  DAYS DESRD     TONS OH  TONS REQRD')
006  C
006  C      SORT MDF2 BY RDD
006            CALL SOPEN3($50,$70,63,10,KEY)
006  50        READ(26,END=60) MDF2
```

```
006          CALL SRREL(MDF2,63)
006          GO TO 50
006   60     CALL SSORT
006   C
006   C    FIRST RECORD SHOULD BE SEQ NOS.
006   70     CALL SRRET(MDF2,63,$1000)
006          DO 90 I = 1,7
006   90     SEQ(I) = MDF2(I)
006   C
006   C    WRITE CARDS FOR THEATER BASE
006          AVL = 999
006          RDD = 0
006          DO 96 K = 1,NTH
006          DO 95 I = 1,3
006          IF(XRS(I,K) .EQ. 0.) GO TO 95
006          SEQ(K) = SEQ(K) + 1
006          WRITE(24,497) SEQ(K),PLANID(K),RT(I),RD(I),RN(1,I+5),RN(2,I+5),
006         * LVL(I),MD(4),AVL,RDD,DEST(1,K),DEST(1,K),DEST(1,K),IZERO
006          IF(I .EQ. 3) GO TO 94
006          AMT = XRS(I,K) + .5
006          WRITE(24,498) SEQ(K),PLANID(K),AMT
006          GO TO 95
006   94     AMT = XRS(I,K) / 1000. + .5
006          WRITE(24,499) SEQ(K),PLANID(K),AMT
006   C    CONVERT BARRELS TO TONS
006          XRS(I,K) = XRS(I,K) / 7.213
006   95     CONTINUE
006   96     CONTINUE
006   C
006          GO TO 399
006   C  MAIN LOOP
006   C
006   200    DAY=DAY+1
006   C    THEND = 0 - THEATER IS CURRENT
006   C    THEND = 1 - THEATER HAS ENDED. NEEDS DATA WRITTEN
006   C    THEND = 2 - THEATER HAS ENDED. ALL DATA WRITTEN
006          DO 201 K = 1,NTH
006          IF(THEND(K) .NE. 0) GO TO 201
006          PR = PER(K)
006          IF(DAY .GT. FRD(PR,K)) PER(K) = PER(K) + 1
006          IF(DAY .LE. LDAY(K)) GO TO 201
006          PER(K) = PER(K) - 1
006          THEND(K) = 1
006   201    CONTINUE
006          DO 202 K = 1,NTH
006          IF(THEND(K) .NE. 2) GO TO 206
006   202    CONTINUE
006          GO TO 900
006   C
006   C    COMPUTE FILLER IF IT'S TIME
006   206    DO 207  K =  1,NTH
006          IF(DAY .NE. CDAY(K)) GO TO 207
006          XFR(K) = XFR(K) * XSTREN(K)
006          PRINT 410, DAY,K,KU(K),XSTREN(K)
006          LNS = LNS + 1
006   207    CONTINUE
006   C  SET ANCHOR POINT FOR BUILDUP FUNCTION
```

```
006   209    DO 220 K = 1,NTH
006            IF(DAY.NE.RDAY(K)) GO TO 220
009            IF(KU(K) .EQ. 0) GO TO 215
006            IF(THEND(K).NE. 0) GO TO 220
006            PR = PER(K)
006            DO 210 I=1,3
006   C      DAYS OF SUPPLY ON HAND AT END OF RDAY-1
006            XRLEV(I,K) = XRS(I,K)/(XCNSM(I,K,PR)*XSTREN(K))
006            XSLOPE(I,K) = (XLEVEL(I,K) - XRLEV(I,K))/(BLDUP(K) - RDAY(K))
006   C      ALREADY OVER REQUIRED LEVEL?
006            IF(XSLOPE(I,K) .GT. 0.) GO TO 210
006            XSLOPE(I,K) = 0
006            XRLEV(I,K) = XLEVEL(I,K)
006   210    CONTINUE
009            GO TO 220
009   C      NO UNITS ON RDAY. UP IT TO NEXT DAY
009   215    RDAY(K) = RDAY(K) + 1
006   220    CONTINUE
006   C
006   300    CONTINUE
006            IF(RDD.GT.DAY) GO TO 400
006   C
006   C
006   C    ENTER NEXT FORCE UNIT DATA
006            DO 301 ITH = 1,IN
006            IF(THR .EQ. THTR(ITH)) GO TO 302
006   301    CONTINUE
006            PRINT 2000, THTR,SRC,UIC,TPSN,THR
006   2000   FORMAT(7I6/' NO THEATER MATCH FOR ',2A6,1X,A6,1X,I5,1X,I3)
006            LNS = LNS + 2
006            GO TO 399
006   302    ITH = THTRID(ITH)
006            IF(THEND(ITH) .NE. 0) GO TO 399
006            XPAX = PAX
006            XSTREN(ITH)=XSTREN(ITH)+XPAX
006            IF(KU(ITH) .EQ. 0) FDAY(ITH) = DAY
006            KU(ITH)=KU(ITH)+1
006            IF(DAY .LE. CDAY(ITH)) GO TO 399
006            POM = 1
006            IF(MCODE .EQ. 1) POM = 2
006   C
006   C    ACCOMPANYING SUPPLY
006            DO 305 I=1,3
006   C      TOTAL TONS ON HAND
006   305    XRS(I,ITH)=XRS(I,ITH)+XASR(I,ITH,POM)*XPAX
006   C
006   C
006   C    READ NEXT UNIT
006   399    CALL SRRET(MDF2,63,$1000)
006   C
006            GO TO 300
006   C
006   C
006   C
006   C    CHECK 5-DAY INTERVAL
006   400    IF(DAY/5*5.NE.DAY) GO TO 600
006   C
```

```
006            JDAY = DAY - 4
006      C  WRITE 5-DAY ROLLUPS
006      C
006      C  CYCLE ON THEATER
006            DO 520 K = 1,NTH
006            IF(THEND(K) .EQ. 2) GO TO 520
006            IF(THEND(K) .EQ. 1) THEND(K) = 2
006      C
006      C  CHECK FOR DAY TO COMMENCE RESUPPLY
006            IF(DAY.LT.RDAY(K)) GO TO 520
006            IDAY = JDAY
006            IF(IDAY .LT. FDAY(K)) IDAY = FDAY(K)
006            IF(IDAY .GT. ADAY(K)) MOE(K) = MD(2)
006            AVL = AVDAY(K)
006            IF(AVL .LT. 0) AVL = IDAY + AVL
010            IF(AVL .GE. IDAY) AVL = CDAY(K)
006      C
006            SMOB(1) = IDAY
006            SMOB(2) = PLANID(K)
006            SMOB(3) = DEST(2,K)
006            SMOB(7) = MOE(K)
006            SMOB(8) = AVL
006            PR = PER(K)
006            XDAY = DAY - RDAY(K)
006      C
006      C   CYCLE ON SUPPLY TYPE
006      C
006            IF(LNS .LT. 50) GO TO 403
006            PRINT 10
006            LNS = 1
006      403   DO 510 I = 1,3
006      C  SET RESUPPLY REQUIREMENTS FOR TYPE
006      C  XDESIR = NEEDED - DAYS OF SUPPLY
006            XDESIR = XSLOPE(I,K) * XDAY + XRLEV(I,K)
006            IF(DAY .EQ. RDAY(K)) XDESIR = XRLEV(I,K)
006            IF(DAY.GE.BLDUP(K)) XDESIR=XLEVEL(I,K)
006      C  XDESRD = NEEDED QUANTITY OF SUPPLY
006            XDESRD = XDESIR*XCNSM(I,K,PR)*XSTREN(K)
006            XAMT = XDESRD - XRS(I,K)
006            IF(XAMT .LE. 0.) GO TO 510
006            XOHD = XRS(I,K)/(XCNSM(I,K,PR)*XSTREN(K))
006      C   CONVERT TO THOUSANDS OF BARRELS
006            IF(I .EQ. 3) XAMT = 7.213 * XAMT /1000.
006            XRS(I,K)=XDESRD
006            LNS = LNS + 1
006            PRINT 410, DAY,K,KU(K),XSTREN(K),XRPLCE(K),I,XOHD,XDESIR,XRS(I,K),
006           * XAMT
006      410   FORMAT(1X,I3,3X,I1,3X,I5,2(2X,F10.0),5X,I1,2X,4(2X,F10.0))
006      C
006      C
006               IST = 1
006               IND = 18
006               IF(I .EQ. 3) IST = 19
006               IF(I .EQ. 3) IND = 19
006      C  CYCLE ON ORIGIN
006            DO 500 J = IST,IND
006      C
```

```
006              IF(I .EQ. 3) GO TO 450
006              IF(XPCT(I,J) .EQ. 0.) GO TO 500
006              AMT = XAMT * XPCT(I,J) + .5
006                  GO TO 460
006       450        AMT = XAMT + .5
006       C
006       460    AMT1 = AMT - 999999
006              IF(AMT1 .GT. 0) AMT = 999999
006              IF(AMT .GT. 12 .OR. I .EQ. 3) GO TO 480
006       C    SEND MINIMUM OF 12 SHORT TONS
006              XRS(I,K) = XRS(I,K) + FLOAT(12 - AMT)
006              AMT = 12
006       C
006       C   WRITE MORSA CARDS
006       480    SEQ(K) = SEQ(K) + 1
006              WRITE(24,497) SEQ(K),PLANID(K),RT(I),RD(I),RN(1,I),RN(2,I),LVL(I),
006            *   MOE(K),AVL,IDAY,GORG(J),DEST(1,K),DEST(1,K),IZERO
006       497    FORMAT(J4,'A',A1,'A     ',2A1,2A6,A3,A1,2X,J3,J3,6X,3A4,J6,19X)
006       C    BULK/AMMO
006              IF(I .NE. 3) WRITE(24,498) SEQ(K),PLANID(K),AMT
006       498    FORMAT(J4,'A',A1,'W',6X,J6,61X)
006       C    POL
006              IF(I .EQ. 3) WRITE(24,499) SEQ(K),PLANID(K),AMT
006       499    FORMAT(J4,'A',A1,'W',12X,J6,55X)
006              KM=KM+2
006              SMOB(4) = I
006              SMOB(5) = AMT
006              SMOB(6) = NORG(J)
006              WRITE(23) SMOB
006              WRITE(12,496) AVL,RT(I),RD(I),SEQ(K),GORG(J),DEST(1,K),MOE(K),IDAY
006            *   ,DEST(2,K),NORG(J),LVL(I),RN(1,I),RN(2,I),AMT,I
006       496    FORMAT(3X,J3,2A1,4X,5X,J6,2A4,A1,4X,1X,3J3,2X,5X,A3,2A6,4X,2J6)
006              IF(AMT1 .LE. 0) GO TO 500
006              AMT = AMT1
006              GO TO 460
006       500    CONTINUE
006       510    CONTINUE
006       C
006       C   REPLACEMENTS
006       C
006       C   ENTER NOPAX
006              IRPL = XRPLCE(K) + .5
006              IF(IRPL .EQ. 0) GO TO 520
006              SEQ(K) = SEQ(K) + 1
006              XRPLCE(K) = 0.
006       C
006              WRITE(24,497) SEQ(K),PLANID(K),RT(1),RD(1),RN(1,4),RN(2,4),LVL(1),
006            *   MD(3),AVL,IDAY,GORG(20),DEST(1,K),DEST(1,K),IRPL
006              KM=KM+1
006              SMOB(4) = 4
006              SMOB(5) = IRPL
006              SMOB(6) = NORG(20)
006              SMOB(7) = MD(3)
006              WRITE(23) SMOB
006              WRITE(12,496) AVL,RT(1),RD(1),SEQ(K),GORG(20),DEST(1,K),MD(3),IDAY
006            *   ,DEST(2,K),NORG(20),LVL(1),RN(1,4),RN(2,4),IRPL,SMOB(4)
006       520    CONTINUE
```

```
006   C
006   C
006   C
006   C  DAILY COMPUTATIONS
006   600    CONTINUE
             SMOB(1) = DAY
006   C  CONSUMPTION
006          DO 700 K = 1,NTH
006          IF(KU(K) .EQ. 0) GO TO 700
006          IF(THEND(K) .NE. 0) GO TO 700
006          PR = PER(K)
006          IF(IDAY .GT. ADAY(K)) MOE(K) = MD(2)
006          AVL = AVDAY(K)
006          IF(AVL .LT. 0) AVL = DAY + AVL
010          IF(AVL .GE. DAY) AVL = CDAY(K)
006          SMOB(2) = PLANID(2)
006          SMOB(3) = DEST(2,K)
006          SMOB(7) = MOE(K)
006          SMOB(8) = AVL
006          IF(LNS .LT. 50) GO TO 605
006          PRINT 10
006          LNS = 1
006   605    DO 650 I=1,3
009          IF(XCNSM(I,K,PR) .EQ. 0.) GO TO 650
006   C    TOTAL LBS ON HAND AFTER CONSUMPTION
006          XRS(I,K) = XRS(I,K)-XSTREN(K)*XCNSM(I,K,PR)
006          IF(DAY .GE. RDAY(K)) GO TO 650
006   C
006   C    CHECK FOR EMERGENCY RESUPPY SHIPMENT
006          XOHD = XRS(I,K)/(XCNSM(I,K,PR) * XSTREN(K))
006          IF(XOHD .GE. XSAFE(I,K)) GO TO 650
006   C    GENERATE RESUPPLY TO BRING UP TO SAFE
006          X = XSAFE(I,K) * XCNSM(I,K,PR) * XSTREN(K)
006          XAMT = X - XRS(I,K)
006          IF(XAMT .EQ. 0.) GO TO 650
006          IF(I .EQ. 3) XAMT = 7.213 * XAMT / 1000.
006          PRINT 410, DAY,K,KU(K),XSTREN(K),XRPLCE(K),I,XOHD,XSAFE(I,K),
006         * XRS(I,K),XAMT
006          LNS = LNS + 1
006          XRS(I,K) = X
006   C  CYCLE ON ORIGIN
006              IST = 1
006              IND = 18
006              IF(I .EQ. 3) IST = 19
006              IF(I .EQ. 3) IND = 19
006          DO 630 J = IST,IND
006   C  SEQUENCE NUMBER
006          SEQ(K)=SEQ(K)+1
006   C
006          IF(I .EQ. 3) GO TO 610
006          IF(XPCT(I,J) .EQ. 0.) GO TO 630
006          AMT = XAMT * XPCT(I,J) + .5
006              GO TO 620
006   610        AMT = XAMT + .5
006   C
006   C  WRITE MORSA CARDS
006   620    WRITE(24,497) SEQ(K),PLANID(K),RT(I),RD(I),RN(1,I),RN(2,I),LVL(I),
```

```
006          •    MOE(K),AVL,DAY,GORG(J),DEST(1,K),DEST(1,K),IZERO
006     C    BULK/AMMO
006            IF(I .NE. 3) WRITE(24,498) SEQ(K),PLANID(K),AMT
006     C    POL
006            IF(I .EQ. 3) WRITE(24,499) SEQ(K),PLANID(K),AMT
006            KM=KM+2
006            N = I
006            SMOB(4) = I
006            SMOB(5) = AMT
006            SMOB(6) = NORG(J)
006            WRITE(23) SMOB
006            WRITE(12,496) AVL,RT(I),RD(I),SEQ(K),GORG(J),DEST(1,K),MOE(K),DAY,
006          •    DEST(2,K),NORG(J),LVL(1),RN(1,1),RN(2,1),AMT,I
006     630    CONTINUE
006     650    CONTINUE
006     C    CHECK FOR FILLER
006            IF(DAY .NE. BDAY(1,K)) GO TO 690
006     C    SEND FILLER
006            FIL = XFR(K) * XBFR(1,K) + .5
006            BDAY(1,K) = BDAY(2,K)
006            BDAY(2,K) = BDAY(3,K)
006            BDAY(3,K) = 99999
006            XBFR(1,K) = XBFR(2,K)
006            XBFR(2,K) = XBFR(3,K)
006            SEQ(K) = SEQ(K) + 1
006            WRITE(24,497) SEQ(K),PLANID(K),RT(1),RD(4),RN(1,5),RN(2,5),LVL(1),
006          •    MD(3),AVL,DAY,GORG(20),DEST(1,K),DEST(1,K),FIL
006            KM = KM + 1
006            SMOB(7) = MD(3)
006            SMOB(4) = 4
006            SMOB(5) = FIL
006            SMOB(6) = NORG(20)
006            WRITE(23) SMOB
006            WRITE(12,496) AVL,RT(1),RD(4),SEQ(K),GORG(20),DEST(1,K),MD(3),DAY,
006          •    DEST(2,K),NORG(20),LVL(1),RN(1,5),RN(2,5),FIL,SMOB(4)
006     C    REPLACEMENTS
006     C    CHECK FOR D-DAY
006     690    IF(DAY.LT.DDAY(K)) GO TO 700
006            XADD = XSTREN(K)*XRPLC(PR,K)/1000.
006            XRPLCE(K) = XRPLCE(K) + XADD
006     700    CONTINUE
006     C
006            GO TO 200
006     C
006     C
006     C
006     C
006     C    PRINT STATISTICS
006     900    CONTINUE
006            ENDFILE 24
006            ENDFILE 23
006            ENDFILE 12
006            REWIND 23
006            IF(ENDM) GO TO 909
006            PRINT 901
006     901    FORMAT(' UNITS NOT PROCESSED ')
006     905    CALL SRRET(MDF2,63,$909)
```

```
006         PRINT 902, SRC,UIC,TPSN,THR,RDD
006   902   FORMAT(1X,2A6,1X,A6,3(1X,I5))
006         GO TO 905
006   909   CALL SMOBST
006         PRINT 910,KU,KM,SEQ
006   910   FORMAT(' UNITS IN THEATER         ', 7I7/' CARDS PRODUCED ', I7/
006      *        ' FINAL SEQUENCE NUMBERS ', 7I7)
006         STOP
006   C
006   C   RAN OUT OF MDF2 BEFORE ALL PERIODS SATISFIED
006   1000  PRINT 1010, DAY
006   1010  FORMAT(' END OF MDF2 FILE. DAY = ',I5)
006         RDD = 999999
006         ENDM = .TRUE.
006         GO TO 400
006         END

END ELT.

@HDG                MRG PROGRAM MEGEMORSA        **UNCLASSIFIED** .L,0
```

```
@ELT,L    58CAMP.MERGEMORSA
ELT007 RL1870 08/16-07:43:47-(0,)
   000             IMPLICIT INTEGER (A-Z)
   000                 DIMENSION IN(14),SEQ(14),SVC(14),PLAN(14),INN(13)
   000                 EQUIVALENCE (IN(2),INN(1))
   000                 DATA SVC/14*'A'/
   000      C     MERGES UNIT AND NON-UNIT MORSA CARDS ON ONE FILE
   000                 IGO = 1
   000      C     UNIT CARDS
   000      10         READ(10,1,END=19) IPLN,INN
   000      1          FORMAT(5X,A1,12A6,A2)
   000                 GO TO 100
   000      C     READ NON-UNIT CARDS
   000      19         IGO = 2
   000      20         READ(24,1,END=200) IPLN,INN
   000      C     FIX UP SEQ NO. IF NECESSARY
   000      C     CARD TYPE A?
   000      100        IF(FLD(0,6,IN(2)) .NE. '@@@@@A') GO TO 140
   000                 DO 110 I = 1,14
   000                 IF(PLAN(I) .EQ. IPLN) GO TO 130
   000                 IF(PLAN(I) .EQ. 0) GO TO 125
   000      110        CONTINUE
   000      125        PLAN(I) = IPLN
   000      130        SEQ(I) = SEQ(I) + 1
   000                 IF(SEQ(I) .LT. 10000) GO TO 140
   000      C     OVER 4 CHAR, START OVER WITH SVC CODE = B
   000                 SEQ(I) = 1
   000                 SVC(I) = 'B'
   000      C     FIX UP WORD ONE
   000      140        ENCODE(6,150,IN(1)) SEQ(I),SVC(I),IPLN
   000      150        FORMAT(J4,A1,A1)
   000                 CALL NTRAN(8,1,14,IN,L,22)
   000                 GO TO (10,20), IGO
   000      C     WRITE END OF FILE
   000      200        CALL NTRAN(8,9)
   000                 PRINT 210,PLAN,SVC,SEQ
   000      210        FORMAT(1X,'PLANID',5X,14A6/1X,'SERVICE',4X,14A6/1X,'SEQUENCE'
   000            *  ,3X,14(J4,2X))
   000                 STOP
   000                 END

END ELT.
```

@HDG   MRG PROGRAM PCKAGE   **UNCLASSIFIED** .L,0

```
@ELT,L   58CAMP.PCKAGE
ELT007 RLIB70 08/16-07:43:49-(5,)
001  C   CREATES SMOBSMOD PACKAGES OF MDF2 AND NUR OUTPUT
001  C   INPUT FILES  26 - MDF2, 23 - SMOB OUTPUT FROM NUR (SORTED)
001  C   OUTPUT FILES  22 - SMOBSMOD CARDS
001         IMPLICIT INTEGER (A-Y)
001         DIMENSION MDF2(63),KEY(31)
001         DIMENSION SRC(2),MDF1(16)
001         EQUIVALENCE  (MDF1( 1),SRC),        @ 2A6
001        •             (MDF1( 3),UIC),        @  A6
001        •             (MDF1( 4),TPSN),       @  I5
001        •             (MDF1( 8),ONOD),       @  I3
001        •             (MDF1( 9),DNOD),       @  I3
001        •             (MDF1(10),RDD ),       @  I3
001        •             (MDF1(11),AVL ),       @  I3
001        •             (MDF1(12),MODE),       @  A1
005        •             (MDF1(13),THR ),       @  I1
001        •             (MDF1(14),MCODE)       @  I1
001         DIMENSION TUCHA(22,2)
001         EQUIVALENCE                             (TUCHA( 5,2),    PAX),
001        •           (TUCHA( 6,1),ZVEHNA),
001        •           (TUCHA( 7,1),ZNSDNA),    (TUCHA( 7,2),NSDNAT),
001        •           (TUCHA( 8,1),ZNSDOU),    (TUCHA( 8,2),NSDOUT),
001        •           (TUCHA( 9,1),ZVEHOU),
001        •           (TUCHA(10,1),ZNVOUT),
001        •           (TUCHA(11,1),ZNSOVC),    (TUCHA(11,2),NSDOVC),
001        •           (TUCHA(12,1),ZNSOVN),    (TUCHA(12,2),NSDOVN),
001        •           (TUCHA(13,1),ZVHOVC),
001        •           (TUCHA(14,1),ZVHOVN),
001        •           (TUCHA(15,1),ZNVOVN),
001        •           (TUCHA(16,1),ZNVOVC),
001        •           (TUCHA(17,1),ZVEHBC),
001        •           (TUCHA(18,1), ZNVBC)
001         EQUIVALENCE (MDF2,MDF1), (MDF2(18),TUCHA(1,1)),
001        •            (MDF2(62),ASUPLY), (MDF2(63),ACAMMO)
001         DATA KEY/ 9,1,36,1,0,1,        @ DNOD
001        •         12,1,36,1,0,2,        @ MODE (MODIFIED)
001        •          8,1,36,1,0,3,        @ ONOD
001        •         10,1,36,1,0,4,        @ RDD
001        •         11,1,36,1,0,5,        @ AVL
001        •         99999/
001         LOGICAL ENDM, ENDS
001         DIMENSION SMOB(8)
001  C   1 = RDD, 2 = PLANID, 3 = DEST, 4 = TYPE, 5 = AMT, 6 = ORIG, 7 = MODE,
001  C   8 = AVL
001         DIMENSION ZTUCHA(22,2)
001         EQUIVALENCE (TUCHA,ZTUCHA)
001         DIMENSION STOT(17,3),ZSTOT(17,3),MTOT(17),ZMTOT(17)
001         EQUIVALENCE (STOT,ZSTOT), (MTOT,ZMTOT)
001         DIMENSION IN(24),ZIN(24),KEY1(37)
001         EQUIVALENCE (IN,ZIN)
001         DATA KEY1/ 2,1,36,1,0,1,   @ RDD
001        •           3,1,36,1,0,2,   @ AVL
001        •           4,1,36,1,0,3,   @ DEST
001        •           5,1,36,1,0,4,   @ ORIG
001        •           1,1,36,1,0,5,   @ PCK
001        •           6,1,36,1,0,6,   @ MODE
```

```
001          *          99999/
001            DIMENSION PACK(4,600),KPCK(3), MD1(4),MD2(4), MINTP(2)
001            DATA MD1 /1,0,2,3/, MD2 /2,1,3,4/, MINTP(2)/100000/
003            DIMENSION XTHR(7), XDAY(7)
001     C
001     C
001            WRITE(21,1)
001     1      FORMAT(5X,'SRC',9X,'UIC',4X,'TPSN    PACK    DEST    ORIG    MODE   '
001          * 'RDD    AVL    UNITS   TOT PAX     TOT STON     TOT SQ FT')
001            WRITE(20,2)
001     2      FORMAT('    PACK    DEST    ORIG    MODE  RDD    AVL   # UNIT',
001          *   '  TOT PAX     TOT STON     SQ FT/BRL')
001     C
001     C    SORT MDF2
001     C
001            CALL SOPEN3($10,$50,63,12,KEY)
001     10     READ(26,END=30) MDF2
001     C    SKIP IF MORSA SEQ NUMBERS
001            IF(RDD .EQ. -99999) GO TO 10
001     C    SKIP IF NO MOVE
001            IF(MCODE .EQ. 0) GO TO 10
001     C    FIX UP MODE FOR SORT
001            M = 2
001            IF(MODE .EQ. 'A') M = 1
001            IF(MODE .EQ. 'S') M = 4
001            IF(ZVEHNA + ZNSDNA .NE. 0.) M = 4
001            T = 100000
001            IF(TPSN .LT. 20000) T = TPSN
001            MODE = T*10 + M
001            CALL SRREL(MDF2,63)
001            GO TO 10
001     C
001     30     READ(5,31) DTRDD,DTAVL
001     31      FORMAT(I2,1X,I2)
001            PRINT 32, DTRDD,DTAVL
001     32     FORMAT(' DELTA FOR RDD = ',I2/
001          *          ' DELTA FOR AVL = ',I2)
003            READ(5,33) (XTHR(I),XDAY(I),I=1,7)
003     33     FORMAT(7(I1,1X,I3,1X))
003            PRINT 34, XTHR,XDAY
003     34     FORMAT(' THEATER ',7I6/' DAY',5X,7I6)
001     C
001            CALL SSORT
001     50     CALL SRRET(MDF2,63,$500)
001            KU = 1
001            GO TO 200
001     C
001     100    IF(ONOD.NE.M8 .OR. DNOD.NE.M9 .OR. MODE.NE.MCOMP .OR.
001          *       RDD.GT.LIMRDD .OR. AVL.GT.LIMAVL) GO TO 250
001     150    IF(AVL .GT. AVLSAV) AVLSAV = AVL
001     C    ADD TO PACKAGE - UNIT DATA
001            DO 160 I = 1,14
001     C    NSDA? USE SQ FT
001            GO TO (155,151,155,155,151,151,155,155,151,151,151,151,151,151),I
001     151    ZMTOT(I) =  ZMTOT(I) + ZTUCHA(I+4,1)
001            IF(ZMTOT(I) .GT. 99999999.) GO TO 170
001            GO TO 160
```

```
001   155    MTOT(I) = MTOT(I) + TUCHA(I+4,2)
001          IF(MTOT(I) .GT. 99999999) GO TO 170
001   160    CONTINUE
001          GO TO 190
001   C
001   C   SUBTRACT FROM OVERFILLED PACKAGE
001   170    DO 180 J = 1,I
001          GO TO (175,171,175,175,171,171,175,175,171,171,171,171,171,171),I
001   171    ZMTOT(I) =  ZMTOT(I) - ZTUCHA(I+4,1)
001          GO TO 180
001   175    MTOT(I) = MTOT(I) - TUCHA(I+4,2)
001   180    CONTINUE
001          PRINT 181, I,ZMTOT(I),MTOT(I),ZTUCHA(I+4,1),
001         •  ZTUCHA(I+4,2),ZADD
001   181    FORMAT(' MDF ',2I5,1X,F11.2,1X,I9,1X,F11.2,1X,I9,1X,F11.2)
001          IGO = 1
001          GO TO 250
001   C
002   190    ZADD = ACAMMO + ASUPLY
001          I = 14
001          IF(ZADD + ZMTOT(14) .GT. 99999999.) GO TO 170
001          ZMTOT(14) = ZMTOT(14) + ZADD
001   C
001   C   SAVE INFO ABOUT PACKAGE
001          NPCK = NPCK + 1
001          K = NPCK
001          DO 195 I = 1,4
001   195    PACK(I,K) = MDF2(I)
001          MINTP(1) = MINO(MINTP(1),TPSN)
001          IF(K .LT. 600) GO TO 199
001          IGO = 1
001          JGO = 1
001          GO TO 250
001   C
001   C GET NEXT RECORD
001   199    CALL SRRET(MDF2,63,$500)
001          KU = KU + 1
001          GO TO 100
001   C
001   C   SAVE DATA FROM PACKAGE'S STARTING DATA
001   200    LIMRDD = RDD + DTRDD
001          LIMAVL = AVL + DTAVL
001          ZRDD = RDD
001          AVLSAV = AVL
001          M8 = ONOD
001          M9 = DNOD
001          MCOMP = MODE
001          M12 = MODE - 1000000
001          IF(TPSN .LT. 20000) M12 = MODE - TPSN*10
001          MINTP(1) = 100000
001          CBT = 2
001          IF(TPSN .LT. 20000) CBT = 1
003          RDDSAV = RDD
003          THRSAV = THR
001          GO TO 150
001   C
001   250    CONTINUE
```

```
001   C     WRITE OUT PACKAGE  - UNIT DATA
001           ZAVL = AVLSAV
001           PCK = PCK + 1
001           ZTOTST = 0
001           ZTOTSF = 0
001           DO 260 I = 1,14
001           GO TO (255,251,255,255,251,251,255,255,251,251,251,251,251,251),I
001   251     ZMTOT(I) = ZMTOT(I) / 1000.
001           ZTOTST = ZTOTST + ZMTOT(I)
001           GO TO 260
001   255     ZMTOT(I) = FLOAT(MTOT(I)) / 1000.
001           IF(I .EQ. 1) GO TO 260
001           ZTOTSF = ZTOTSF + ZMTOT(I)
001   260     CONTINUE
001           DO 270 I = 15,7,-1
001   270     ZMTOT(I) = ZMTOT(I-1)
001           ZMTOT(6) = 0
003   C     CHANGE PACKAGE MODE IF NECESSARY
003           DO 271 I = 1,7
004           IF(THRSAV .EQ. XTHR(I)) GO TO 273
003   271     CONTINUE
003           PRINT 272, THRSAV
003   272     FORMAT(' ERROR IN THEATER INPUT. NEED ENTRY FOR THEATER',I5)
003           RETURN 0
003   273     IF(RDDSAV .LT. XDAY(I)) GO TO 279
003           M12 = 4
003   279     WRITE(22) PCK,ZRDD,ZAVL,M9,M8,M12,(ZMTOT(I),I=1,17),MINTP(CBT)
001   C     ZERO PACKAGE DATA
001           IF(NPCK .GT. MAX) MAX = NPCK
001           WRITE(21,281) (((PACK(I,J),I=1,4),PCK,M9,M8,MD1(M12),ZRDD,
001          •  ZAVL,NPCK,ZMTOT(1),ZTOTST,ZTOTSF),J=1,NPCK)
001   281     FORMAT((1X,2A6,2X,A6,5(2X,I5),2(2X,F4.0),2X,I5,3(2X,F9.3)))
001           WRITE(20,422) PCK,M9,M8,MD1(M12),ZRDD,ZAVL,NPCK,ZMTOT(1),ZTOTST,
001          •  ZTOTSF,MINTP(CBT)
001           NPCK = 0
001           DO 280 I = 1,14
001   280     MTOT(I) = 0
001           IF(IGO .EQ. 0) GO TO 299
001           IGO = 0
001           MINTP(1) = TPSN
001           IF(JGO .EQ. 0) GO TO 150
001           JGO = 0
001           MINTP(1) = 100000
001           GO TO 199
001   299     IF(.NOT. ENDM) GO TO 200
001   C     READ FIRST SMOB RECORD
001           READ(23) SMOB
001           KV = 1
001           GO TO 510
001   C
001   C     READ NUR DATA
001   300     IF(SMOB(6).NE.S6 .OR. SMOB(3).NE.S3
001          •    .OR. SMOB(1).GT.LIMRDD .OR. SMOB(8).GT.LIMAVL) GO TO 400
001   C     ADD TO PACKAGE - NUR DATA
001   C     1 = DRY BULK (TONS), 2 = AMMO (TONS), 3 = POL (1000 BARRELS), 4 = PEOPLE
001   C      OUTPUT ALL IN THOUSANDS
001   C
```

```
001   360    M = 2
001          IF(SMOB(7) .EQ. 'A') M = 1
001          IF(SMOB(7) .EQ. 'S') M = 3
001          IF(SMOB(8) .GT. AVLSAV) AVLSAV = SMOB(8)
001          I = SMOB(4)
001          STOT(I,M) = STOT(I,M) + SMOB(5)
001          IF(I .EQ. 3 .AND. STOT(I,M) .GT. 99999) GO TO 362
001          IF(STOT(1,M) .LE. 99999999) GO TO 390
001   C
001   C    SUBTRACT FROM OVERFILLED PACKAGE
001   362    STOT(I,M) = STOT(I,M) - SMOB(5)
001          IGO = M
001          PRINT 361, I,M,STOT(I,M),SMOB(5),S6,S3
001   361    FORMAT(' SMOB ',6I10)
001          GO TO 400
001   C
001   390    KPCK(M) = KPCK(M) + 1
001   C
001          READ(23,END=550) SMOB
001          KV = KV + 1
001          GO TO 300
001   C
001   400    CONTINUE
001   C    WRITE OUT PACKAGE - NUR DATA
001          ZAVL = AVLSAV
001          IS = 1
001          IE = 3
001          IF(IGO .NE. 0) IS = IGO
001          IF(IGO .NE. 0) IE = IGO
001          DO 435 M = IS,IE
001          IF(KPCK(M) .EQ. 0) GO TO 435
001          PCK = PCK + 1
001          MPCK = MPCK - 1
001          ZSTOT( 6,M) = STOT(3,M)                    @ POL
001          ZSTOT(16,M) = FLOAT(STOT(1,M))/1000.       @ BULK
001          ZSTOT(17,M) = FLOAT(STOT(2,M))/1000.       @ AMMO
001          ZSTOT( 1,M) = FLOAT(STOT(4,M))/1000.       @ PEOPLE
001          ZTOTST = ZSTOT(16,M) + ZSTOT(17,M)
001   C    ZERO PACKAGE DATA
001          DO 430 I = 2,4
001   430    STOT(I,M) = 0
001          WRITE(22) PCK,ZRDD,ZAVL,S3,S6,M,(ZSTOT(I,M),I=1,17),MPCK
001          WRITE(20,422) PCK,S3,S6,MDI(M),ZRDD,ZAVL,KPCK(M),ZSTOT(1,M),
001         •   ZTOTST,ZSTOT(6,M),MPCK
001          WRITE(19,422) PCK,S3,S6,MDI(M),ZRDD,ZAVL,KPCK(M),ZSTOT(1,M),
001         •   ZTOTST,ZSTOT(6,M)
001   422    FORMAT(4(2X,I5),2(2X,F4.0),2X,I5,3(2X,F9.3),2X,I5)
001          KPCK(M) = 0
001          STOT(1,M) = 0
001   435    CONTINUE
001          IF(IGO .EQ. 0) GO TO 440
001          IGO = 0
001          GO TO 360
001   440    IF(ENDS) GO TO 1000
001          GO TO 510
001   C
001   500    ENDM= .TRUE.
```

```
001                GO TO 250
001          C
001          C     SAVE DATA FROM PACKAGE'S STARTING RECORD
001          510     S6 = SMOB(6)
001                  S3 = SMOB(3)
001                  ZRDD = SMOB(1)
001                  LIMRDD = SMOB(1) + DTRDD
001                  LIMAVL = SMOB(8) + DTAVL
001                  AVLSAV = SMOB(8)
001                  GO TO 360
001          550     ENDS = .TRUE.
001                  GO TO 400
001          C
001          1000    ENDFILE 22
001                  REWIND 22
001          C
001          C     SORT 22 BY AVL, RDD
001                  CALL SOPEN3($1005,$1030,24,6,KEY1)
001          C     READ OTHER SERVICE DATA, THEN FILE 22
001          1005    READ(18,1006,END=1010) IN
001          1006    FORMAT(I6,2F3.0,2I3,I1,6(F8.3)/11(F8.3),I6)
001          C     FIX UP MODE FOR SORT
001                  J = IN(6) + 1
001                  IN(6) = MD2(J)
001                  KO = KO + 1
001                  PCK = PCK + 1
001                  CALL SRREL(IN,24)
001                  GO TO 1005
001          1010    READ(22,END=1020) IN
001                  CALL SRREL(IN,24)
001                  GO TO 1010
001          1020    REWIND 22
001                  CALL SSORT
001          1030    CALL SRRET(IN,24,$1100)
001                  SEQ = SEQ + 1
001          C     OTHER SERVICE DATA?
001                  IF(IN(24) .EQ. 999999) GO TO 1040
001                  TP = ' '
001                  IF(IN(24).GT.0 .AND. IN(24).LT.100000) ENCODE(5,1029,TP,L) IN(24)
001          1029    FORMAT(J5)
001                  WRITE(22,1031) IN(1),TP,IN(2),IN(3),SEQ
001          1031    FORMAT(' PACKAGE ',J6,5X,A5,11X,'RDD=   ',F6.2,6X,'AVL=   ',F6.2,I6)
001                  IF(IN(24) .LT. 0) GO TO 1050
001          1035    S = SEQ
001                  GO TO 1060
001          C     OTHER SERVICE DATA
001          1040    WRITE(22,1041) (IN(I),I=1,3),SEQ
001          1041    FORMAT(' OTHER SERVICE PACKAGE ',J6,7X,'RDD=   ',F6.2,6X,
001               *  'AVL=   ',F6.2,I6)
001                  GO TO 1035
001          1050    S = -SEQ
001          C     SET MODE PROPERLY FOR CARDS
001          1060    J = IN(6)
001                  IN(6) = MD1(J)
001                  IF(IN(7) .NE. 0) WRITE(22,1032) (IN(I),I=3,7),S
001                  DO 1070 I = 8,23
001                  N = I - 6
```

```
000341    001            IF(IN(I) .NE. 0) WRITE(22,1033) (IN(K),K=3,6),ZIN(I),N,S
000342    001      1070  CONTINUE
000343    001      1032  FORMAT(1X,F9.2,2X,I3,2X,I3,4X,I1,1X,F9.3,14X,'1',I5)
000344    001      1033  FORMAT(1X,F9.2,2X,I3,2X,I3,4X,I1,11X,F9.3,3X,I2,I5)
000345    001      1034  FORMAT(3X,'5444.'22X,'0.',8X,'0.')
000346    001            WRITE(22,1034)
000347    001            GO TO 1030
000348    001      C
000349    001      1100  ENDFILE 22
000350    001            MPCK = -MPCK
000351    001            PRINT 1101, KU,KV,PCK,MPCK,KO,MAX
000352    001      1101  FORMAT(' UNITS PROCESSED ',I7/
000353    001           .        ' NUR   PROCESSED ',I7/
000354    001           .        ' TOT # OF PCKAGES',I7/
000355    001           .        ' # OF NUR  PACKS ',I7/
000356    001           .        ' OTHER SERVICES  ',I7/
000357    001           .        ' MAX UNITS IN PCK',I7)
000358    001            STOP
000359    001            END

END ELT.

@HDG    MRG POST-PROCESSOR PROGRAM INLAY         **UNCLASSIFIED** .L,0
```

```
@ELT,L   58CAMP.LAYIN
ELT007 RL1870 08/16-07:43:54-(25,)
021          IMPLICIT INTEGER(A-Z)
021          DIMENSION F(52), OF(52,24),     @ UNITS 17,15  FAS INPUT, OUTPUT
021      •              U(19),               @ UNIT 11 LOG INPUT
021      •              N(13),               @ UNIT 12 NUR INPUT
021      •              P( 2)                @ UNIT 13 PACKAGE INPUT
021          EQUIVALENCE (N,U), (OF,F)
021          DIMENSION EDATE(2)
021          DIMENSION KEYU(7), KEYP(7), KEYF(7), KEYOF(7)
021          DATA KEYU /2, 1,36,0,0,1,99999/    @ UIC IN UNIT DATA
021          DATA KEYP /1, 1,36,0,0,1,99999/    @ UIC IN PACKAGE DATA
021          DATA KEYF /1,19,36,0,0,1,99999/    @ UIC IN FAS
021          DATA KEYOF/1, 7,90,0,0,1,99999/    @ KEY FIELD IN OUTPUT FAS
021          LOGICAL ENDU, ENDP
021          DATA ZERO/'000000'/
021          DEFINE C01(X) = FLD(0,6,X)
021          DEFINE C02(X) = FLD(6,6,X)
021          DEFINE C03(X) = FLD(12,6,X)
021          DEFINE C04(X) = FLD(18,6,X)
021          DEFINE C05(X) = FLD(24,6,X)
021          DEFINE C06(X) = FLD(30,6,X)
021          DEFINE C12(X) = FLD(0,12,X)
021          DEFINE C13(X) = FLD(0,18,X)
021          DEFINE C14(X) = FLD(0,24,X)
021          DEFINE C15(X) = FLD(0,30,X)
021          DEFINE C23(X) = FLD(6,12,X)
021          DEFINE C24(X) = FLD(6,18,X)
021          DEFINE C25(X) = FLD(6,24,X)
021          DEFINE C26(X) = FLD(6,30,X)
021          DEFINE C34(X) = FLD(12,12,X)
021          DEFINE C35(X) = FLD(12,18,X)
021          DEFINE C36(X) = FLD(12,24,X)
021          DEFINE C45(X) = FLD(18,12,X)
021          DEFINE C46(X) = FLD(18,18,X)
021          DEFINE C56(X) = FLD(24,12,X)
021     C
021     C    SORT UNIT DATA
021     C
021          CALL SOPEN3($10,$30,19,2,KEYU)
021     10   READ(11,1,END=20) U
021     1    FORMAT(19A6)
021          CALL SRREL(U,19)
021          KU = KU + 1
021          GO TO 10
021     20   REWIND 11
021          CALL SSORT
021     30   CALL SRRET(U,19,$40)
021          WRITE(11,1) U
021          GO TO 30
021     40   ENDFILE 11
021          REWIND 11
021     C
021     C    SORT PACKAGE DATA
021     C
021     C    READ HEADER RECORD
021          READ(21,51) P
```

```
021           CALL SOPEN3($50,$70,2,1,KEYP)
021    C    PICK UP UIC AND PACKAGE NUMBER
021    50     READ(21,51,END=60) P
021    51     FORMAT(15X,A6,8X,A6)
021           CALL SRREL(P,2)
021           KP = KP + 1
021           GO TO 50
021    60     CALL SSORT
021    70     CALL SRRET(P,2,$80)
021           WRITE(13,1) P
021           GO TO 70
021    80     ENDFILE 13
021           REWIND 13
021    C
021    C    SORT FAS FILE
021           CALL SOPEN3($100,$120,52,2,KEYF)
021    100    READ(17,END=110) F
021           CALL SRREL(F,52)
021           KF = KF + 1
021           GO TO 100
021    110    CALL SSORT
021    C    NOW PROCESS UNIT RECORDS
021    C
021    120    CALL SRRET(F(1),52,$155)
021           UIC = C46(F(1))*2**18 + C13(F(2))
021           LF = LF + 1
021           READ(11,1) U
021           READ(13,1) P
021           LU = LU + 1
021           LP = LP + 1
021           IF(U(2) - P(1)) 125,130,121
021    121    PRINT 122, U(2),P(1)
021    122    FORMAT(' NO MATCH FOR UNIT ',A6,' ON PACKAGE FILE ',A6)
021           STOP
021    125    PCK = ZERO
021           GO TO 131
021    130    PCK = P(2)
021           READ(13,1) P
021           LP = LP + 1
021    131    IF(U(2) - UIC) 140,160,150
021    C    NO MATCH
021    140    PRINT 141, U(2),UIC
021    141    FORMAT(' NO MATCH FOR UNIT ',A6,' ON FAS ',A6/
021          *     ' JOB IS BEING TERMINATED ')
021           GO TO 515
021    C    NO MATCH. KEEP LOOKING
021    150    WRITE(14) F
021           CALL SRRET(F(1),52,$155)
021           LF = LF + 1
021           IF(ENDU) GO TO 150
021           UIC = C46(F(1))*2**18 + C13(F(2))
021           GO TO 131
021    155    IF(ENDU) GO TO 180
021           PRINT 156, U(2)
021    156    FORMAT(' REACHED END OF FAS BEFORE UNITS ',A6)
021           STOP
021    C    MATCHED FAS. ADD DATA
```

```
025   C     SAVE POSITIONS 37,38 IN 120,121
025   160     C06(F(20)) = C01(F(7))
025           C01(F(21)) = C02(F(7))
025           C46(F( 6)) =      U(1)      @ AVL
021           C01(F( 7)) = '@@@@@U'       @ RT
021           C02(F( 7)) =      U(3)      @ RD
021           C06(F( 8)) =      U(4)      @ MORSA SEQ NO
021                F( 9) =      U(5)      @ MORSA SEQ NO, GEO ORIG
021                F(10) =      U(6)      @ GEO ORIG, DEST
021           C12(F(11)) = C12(U(7))      @ GEO DEST, MODE
021           C26(F(19)) =      U(8)      @ RDD, NODE DEST
021           C14(F(20)) = C14(U(9))      @ NODE DEST, ORIG
023           C15(F(28)) =      PCK       @ PACKAGE NO
021           C06(F(38)) =      U(10)     @ BULK
021           DO 170 I = 11,18
021   170     F(I+28) = U(I)              @ BULK,OVER,OUT,NAT,NSDA,TOTL,SUPLY,PAX,AMMO
021           C01(F(47)) = C01(U(19))     @ AMMO
021   C
021           READ(11,1,END=175) U
021           LU = LU + 1
021           IF(ENDP) GO TO 171
021           IF(U(2) - P(1)) 171,172,121
021   171     PCK = ZERO
021           GO TO 150
021   172     PCK = P(2)
021           READ(13,1,END=176) P
021           LP = LP + 1
021           GO TO 150
021   175     ENDU = .TRUE.
021           GO TO 150
021   176     ENDP = .TRUE.
021           GO TO 171
021   C
021   C     SORT ACCORDING TO KEY FIELD
021   180     ENDFILE 14
021           REWIND 14
021           CALL SOPEN3($185,$195,52,3,KEYOF)
021   185     READ(14,END=190) F
021           CALL SRREL(F,52)
021           GO TO 185
021   190     CALL SSORT
021   195     DO 198 I = 1,24
021           ISAV = I
021           CALL SRRET(OF(1,I),52,$200)
021   198     CONTINUE
021           CALL NTRAN(15,1,1248,OF,L,22)
021           GO TO 195
021   C
021   C     PROCESS NON UNIT DATA - GENERATE NEW FAS RECORDS
021   C
021   C     READ INPUT DATA
021   200     READ(5,211) FICOD,EDATE
021   211     FORMAT(A1,1X,A6,A1)
021           PRINT 212, FICOD,EDATE
021   212     FORMAT(' FICOD = ',A1,'   EDATE = ',A6,A1)
021           F1 = ' '
021           F2 = ' '
```

```
021            F3 = ' '
021            C04(F1) = C01(FICOD)
021            C56(F1) = '@@@@9N'
021            C36(F2) = C13(EDATE(1))
021            C13(F3) =     EDATE(1)
021            C04(F3) = C01(EDATE(2))
021            ICT = ISAV - 1
021      C
021      220   READ(12,221,END=500) N,ITYP
021      221   FORMAT(13A6,I6)
021            ICT = ICT + 1
021            KN = KN + 1
021            DO 225 I = 4,52
021      225   OF(I,ICT) = ' '
021      C
021      C   SET FAS DATA
021      C
021            OF( 2,ICT)  = F2
021            OF( 3,ICT)  = F3
021            OF(12,ICT)  = '    DC'
021            SEQ = SEQ + 1
021            ENCODE(9,226,OF(1,ICT),L) SEQ
021      226   FORMAT(4X,J5)
021            C24(OF( 1,ICT))  = F1
021            DO 227 I = 1,6
021      227   OF(I+ 5,ICT) = N(I)    @ AVL,RT,RD,MORSA SEQ NO, GEO ORG,DEST,MODE
021            DO 228 I = 7,8
021      228   OF(I+12,ICT) = N(I)    @ RDD,NODE DEST,ORIG
021            DO 229 I = 9,12
021      229   OF(I+25,ICT) = N(I)    @ LVL,NAME
021            GO TO (231,231,233,234), ITYP
021      C   DRY/AMMO
021      231   C06(OF(38,ICT)) = C02(N(13))    @ AMT
021            C14(OF(39,ICT)) =     N(13)     @ AMT
021            C15(OF(43,ICT)) =     N(13)     @ TOTAL
021            GO TO 240
021      C   POL
021      233   C56(OF(44,ICT)) = C23(N(13))    @ AMT
021            C13(OF(45,ICT)) =     N(13)     @ AMT
021            GO TO 240
021      C   PAX
022      234   C46(OF(45,ICT)) = C24(N(13))    @ AMT
021            C12(OF(46,ICT)) =     N(13)     @ AMT
021      C
021      240   IF(ICT .NE. 24) GO TO 220
021            CALL NTRAN(15,1,1248,OF,L,22)
021            ICT = 0
021            GO TO 220
021      C
021      500   ICT = ICT + 1
021      550   DO 560 J = ICT,24
021            DO 560 K = 1,52
021      560   OF(K,J) = '999999'
024            CALL NTRAN(15,1,1248,OF,L,9,10,22)
021      515   PRINT 511, KU,LU,KP,LP,KF,LF,KN
021      511   FORMAT(' UNITS READ           ',I6,' PROCESSED ',I6/
021           *       ' PACKAGED UNITS READ ',I6,' PROCESSED ',I6/
```

```
000227      021      *        ' FAS UNITS READ      ',I6,' PROCESSED ',I6/
000228      021      *        ' NON UNIT FAS RECORDS CREATED ',I6)
000229      021        STOP
000230      021        END

END ELT.

@HDG     MRG POST-PROCESSOR PROGRAM PRTPCK          **UNCLASSIFIED** *L,0
```

```
@ELT,L   58CAMP.PRTPCK
ELT007 RL1B70 08/16-07:43:57-(11,)
        007    C    PRINT UNITS IN PACKAGES
        007           IMPLICIT INTEGER (A-Y)
        007           DIMENSION IN(11),ZIN(3),PRT(4,4),JN(7),ZJN(3)
        007    C    READ HEADING RECORD
        007           READ(21,1) IN(1)
        007           PRINT 2
        007    2      FORMAT(1H1,' PACKAGE  DEST  ORIG  MODE  RDD  AVL  UNITS  '
        009         •   'TOT PAX    TOT STON   TOT SQ FT ',22H(1000'S OF PAX, TONS,
        007         •   ' SQ FT)'// 10X,4(9X,'SRC',7X,'UIC',4X,'TPSN'))
        010    10     READ (21,1,END=80) IN,ZIN
        007    1      FORMAT(1X,2A6,2X,A6,5(2X,I5),2X,I3,3X,I3,3X,I5,3(2X,F9.3))
        007           IF(IN(5) .EQ. PACK) GO TO 50
        007           IF(ICT .NE. 0) GO TO 70
        007    15     PACK = IN(5)
        007           ICT = 0
        008           IF(LNS .LT. 52) GO TO 18
        007           PRINT 2
        007           LNS = 0
        007    18     PRINT 3, (IN(I),I=5,11),ZIN
        007    3      FORMAT(/2X,I5,4X,I3,3X,I3,5X,I1,3X,I3,2X,I3,2X,I5,3(2X,F9.3))
        007           LNS = LNS + 2
        007    C
        007    50     ICT = ICT + 1
        007           DO 55 I = 1,4
        007    55     PRT(I,ICT) = IN(I)
        007           IF(ICT .NE. 4) GO TO 10
        007           IF(LNS .LT. 54) GO TO 60
        007           PRINT 2
        007           LNS = 0
        007    60     PRINT 5, PRT
        007    5      FORMAT(10X,4(5X,2A6,1X,A6,1X,I5))
        007           ICT = 0
        007           LNS = LNS + 1
        007           GO TO 10
        007    70     IF(LNS .LT. 54) GO TO 75
        007           PRINT 2
        007           LNS = 0
        007    75     PRINT 5, ((PRT(I,J),I=1,4),J=1,ICT)
        007           ICT = 0
        007           LNS = LNS + 1
        007           GO TO 15
        011    80     IF(ICT .EQ. 0) GO TO 100
        011           IF(LNS .LT. 54) GO TO 85
        011           PRINT 2
        011           LNS = 0
        011    85     PRINT 5, ((PRT(I,J),I=1,4),J=1,ICT)
        011           LNS = LNS + 1
        007    C    PROCESS NUR PACKAGES
        007    100    PRINT 101
        007    101    FORMAT(1H1,' PACKAGE  DEST  ORIG  MODE  RDD  AVL  ENTRY  '
        009         •   'TOT PAX    TOT STON   TOT BARRL ',22H(1000'S OF PAX, TONS,
        007         •   ' BARRELS'//)
        007           LNS = 0
        007    110    READ(19,111,END=1000) JN,ZJN
        007    111     FORMAT(4(2X,I5),2X,I3,3X,I3,3X,I5,3(2X,F9.3))
```

```
000056      008           IF(LNS .LT. 52) GO TO 120
000057      007           LNS = 0
000058      007           PRINT 101
000059      008    120    LNS = LNS + 2
000060      007           PRINT 3,JN,ZJN
000061      007           GO TO 110
000062      007   1000    STOP
000063      007           END

END ELT.

@HDG    MRG POST-PROCESSOR PROGRAM BUILDMRG        ••UNCLASSIFIED•• .L,0
```

```
@ELT,L    24MONTAGNE.BUILD-FASMRG
ELT007 RL1870 08/16-07:44:00-(1,)
   000   010010 IDENTIFICATION DIVISION.
   000   010020 PROGRAM-ID. 'BUILD'.
   000   010030 AUTHOR. J MEYEROWITZ.
   000   010040 DATE WRITTEN. 02 MAY 1973.
   000   010060 REMARKS. READ A FAS FILE FROM TAPE
   000   010070                   307 CHARACTERS BLOCKED 24 PER RECORD.
   000   010080              WRITE A UDF FILE TO MASS STORAGE USING ISFM
   000   010090                   360 CHARACTERS PER RECORD.
   000   020010 ENVIRONMENT DIVISION.
   000   020020 CONFIGURATION SECTION.
   000   020030 SOURCE-COMPUTER. UNIVAC-1108.
   000   020040 OBJECT-COMPUTER. UNIVAC-1108.
   000   020050 INPUT-OUTPUT SECTION.
   000   020060 FILE-CONTROL.
   000   020070      SELECT MASTER-FILE ASSIGN TO MASS-STORAGE FORCE
   000   020080      ACCESS MODE IS RANDOM SYMBOLIC KEY IS REAL-KEY
   000   020085      ORGANIZATION IS INDEXED FILE-DESCRIPTION IS FILE-DESCRIPT.
   000   020120      SELECT FAS-FILE ASSIGN TO UNISERVO FAS-FILE.
   000   020130      SELECT PRINT-FILE ASSIGN TO PRINTER.
   000   030010 DATA DIVISION.
   000   030020 FILE SECTION.
   000   030030 FD   MASTER-FILE
   000   030040      LABEL RECORDS ARE STANDARD
   000   030050      DATA RECORD IS MASTER-REC.
   000   030060 01   MASTER-REC           PICTURE IS X(10752).
   000   031010 FD   FAS-FILE
   000   031020      LABEL RECORD IS OMITTED
   000   031030      RECORD CONTAINS 7488 CHARACTERS
   000   031035      RECORDING MODE IS 1
   000   031040      DATA RECORD IS TAPE-BLOCK.
   000   031050 01   TAPE-BLOCK           PICTURE IS X(7488).
   000   032010 FD   PRINT-FILE
   000   032020      RECORD CONTAINS 132 CHARACTERS
   000   032030      LABEL RECORD IS OMITTED
   000   032040      DATA RECORD IS PRINT-LINE.
   000   032050 01   PRINT-LINE           PICTURE IS X(132).
   000   040010 WORKING-STORAGE SECTION.
   000   040020 77   NDEX                 PICTURE IS 99 COMP SYNC RIGHT VALUE IS 0.
   000   040030 77   N                    PICTURE IS H9      VALUE IS ZERO.
   000   040100 01   WORK-BLOCK.
   000   040110      02   WORK-REC   OCCURS 24 TIMES.
   000   040120           03   PART-1.
   000   040130                04   FILLER PICTURE IS X.
   000   040140                04   WORKID.
   000   040143                  05 FICOD PICTURE IS X.
   000   040144                  05 COMPO PICTURE IS X.
   000   040145                  05 UICCC PICTURE IS X(6).
   000   040146                  05 EDATE PICTURE IS X(6).
   000   040150                04   FILLER PICTURE IS X(11).
   000   040160           03   PART-2.
   000   04162                 04   FILLER     PIC X(12).
   000                    04   THEATER-CODE     PIC X.
   000                    04   FILLER           PIC X(31).
   000                    04   DSCMP    PIC XX.
   000                    04   FILLER   PIC XXX.
```

```
000   04164              04  ESCON      PIC X.
000   04166              04  FILLER     PIC X(108).
000                   04  TRCON           PIC X.
000                   04  FILLER          PIC X(127).
000   050010 01  DATA-REC.
000   050020     02  SEG-1            PICTURE IS X(26).
000   050030     02  FILLER           PICTURE IS X        VALUE IS 'X'.
000   050040     02  SEG-2            PICTURE IS X(333).
000   060010 01  PRT-AREA.
000   060020     02  PRT-ITEM         PICTURE IS X(18)    OCCURS 7 TIMES.
000   080010 01  ER-LINE.
000   080020     02  FILLER         PICTURE IS X(12)   VALUE IS 'INVALID KEY '.
000   080030     02  ER-KEY         PICTURE IS X(14).
000   150010 01  REAL-KEY.
000   150020     02  REC-KEY          PICTURE IS X(14).
000   150030     02  FILLER           PICTURE IS X(4).
000   180010 01  FILE-DESCRIPT.
000   180020     02  NUMBER-OF-RECORDS        PICTURE IS H9(10) VALUE IS 12000.
000   180030     02  RECORD-LENGTH            PICTURE IS H9(10) VALUE IS 360.
000   180040     02  RECORD-LENGTH-MAX        PICTURE IS H9(10) VALUE IS 0.
000   180050     02  RECORD-KEY-LENGTH        PICTURE IS H9(10) VALUE IS 18.
000   180060     02  OVERFLOW-AREA            PICTURE IS H9(10) VALUE IS 10.
000   180070     02  I-O-F                    PICTURE IS H9(10) VALUE IS 1000.
000   180080     02  DATA-NAME-8.
000   180090         03  DATA-NAME-9   PICTURE IS X(3).
000   180100         03  DATA-NAME-10  PICTURE IS X(3).
000   190010 01  INFORM.
000   190020     02  NUMBER-OF-BLOCKS PICTURE IS H9(10).
000   190030     02  NUMBER-OF-INDX-BLOCKS  PICTURE IS H9(10).
000   190040     02  NUMBER-OF-OVERFL-BLOCKS  PICTURE IS H9(10).
000   190050     02  NUMBER-OF-RECORDS          PICTURE IS H9(10).
000   190060     02  NUMBER-OF-RECORDS-IN-IOF  PICTURE IS H9(10).
000   190070     02  NUMBER-OF-RECORDS-DELRTD  PICTURE IS H9(10).
000   190080     02  NUMBER-OF-RECORDS-READ  PICTURE IS H9(10).
000   190090     02  NUMBER-OF-RECORDS-READ-FROM-IF    PICTURE IS H9(10).
000   190100     02  NUMBER-OF-RECORDS-WRITTEN         PICTURE IS H9(10).
000   195010 01  INFORM-FL-DATA.
000   195020     02  NUMBER-OF-BLOCKS PICTURE IS 9(6).
000   195030     02  NUMBER-OF-INDX-BLOCKS  PICTURE IS 9(6).
000   195040     02  NUMBER-OF-OVERFL-BLOCKS  PICTURE IS 9(6).
000   195050     02  NUMBER-OF-RECORDS          PICTURE IS 9(6).
000   195060     02  NUMBER-OF-RECORDS-IN-IOF  PICTURE IS 9(6).
000   195070     02  NUMBER-OF-RECORDS-DELRTD  PICTURE IS 9(6).
000   195080     02  NUMBER-OF-RECORDS-READ  PICTURE IS 9(6).
000   195090     02  NUMBER-OF-RECORDS-READ-FROM-IF    PICTURE IS 9(6).
000   195100     02  NUMBER-OF-RECORDS-WRITTEN         PICTURE IS 9(6).
000   200000 PROCEDURE DIVISION.
000   200010 START-TEST.
000   200020     OPEN INPUT FAS-FILE WITH NO REWIND.
000   200030     OPEN OUTPUT PRINT-FILE.
000   200040     OPEN OUTPUT MASTER-FILE.
000   210000 TAPE-READ.
000   210010     MOVE ZERO TO NDEX.
000   210020     READ FAS-FILE AT END GO TO FINI.
000   210030     MOVE TAPE-BLOCK TO WORK-BLOCK.
000   220000 GET-ITEM.
000   220010     ADD 1 TO NDEX.
```

```
000  220020       IF NDEX = 25 GO TO TAPE-READ.
000  220021        IF WORKID (NDEX) = SPACES GO TO GET-ITEM.
000  220025       IF WORKID (NDEX) = '6666666666666' GO TO GET-ITEM.
000               MOVE THEATER-CODE(NDEX) TO TRCON(NDEX).
000               IF COMPO(NDEX) = '9' MOVE 'DC' TO DSCMP(NDEX).
000  220030       MOVE PART-1 (NDEX) TO SEG-1.
000  220040       MOVE PART-2 (NDEX) TO SEG-2.
000  220050       MOVE WORKID (NDEX) TO REAL-KEY.
000  220055       ADD 1 TO N.
000  220060       IF N = 8 PERFORM LINE-LIST THRU X-LIST.
000  220065       WRITE MASTER-REC FROM DATA-REC INVALID KEY GO TO ERR-0.
000  220070       MOVE REAL-KEY TO PRT-ITEM (N).
000  220080       GO TO GET-ITEM.
000  225010 LINE-LIST.
000  225020       MOVE 1 TO N.
000  225030       WRITE PRINT-LINE FROM PRT-AREA.
000  225040       MOVE SPACES TO PRT-AREA.
000  225050 X-LIST.
000  225060       EXIT.
000  226010 ERR-0.
000  226020       MOVE REAL-KEY TO ER-KEY.
000  226025       WRITE PRINT-LINE FROM ER-LINE.
000  226030       GO TO FINI.
000  230000 FINI.
000  230020       CLOSE MASTER-FILE USING INFORM.
000  230030       MOVE CORRESPONDING INFORM TO INFORM-FL-DATA.
000  230035       WRITE PRINT-LINE FROM PRT-AREA.
000  230040       WRITE PRINT-LINE FROM INFORM-FL-DATA AFTER 2 LINES.
000  230050       CLOSE PRINT-FILE.
001  230010       CLOSE FAS-FILE.
000  230060       STOP RUN.
```

END ELT.

@BRKPT PRINT$

COMPUTER ASSISTED MATCH PROGRAM (CAMP)

CHAPTER VIII

SAMPLE OUTPUT

```
@RUN,/PRNT 8124E,H3883P3043K,UNCLASSIFIED,100,300 . RSORT/TEST

@HDG         RSORT/TEST         ••• UNCLASSIFIED •••

@MSG,N  INPUT FILES

@QUAL UNCLASSIFIED

@ASG,A    •24TESTRQR•     . CAMPSUM, REQUIREMENTS FILE FROM FASTALS

@USE   14,•24TESTRQR•

@ASG,T 15

@ED 24TESTDATA.LOUIN,15.   . UIN TO BE DELETED FROM REQUIREMENTS FILE
CASE UPPER ASSUMED
ED 15.00-08/19/76-10:23:40-(2,)
EDIT
LINES:2 FIELDATA

@MSG,N  OUTPUT FILES

@DELETE,C 24UIN.
FURPUR 27R1 RL71-3 08/19/76 10:23:42

@DELETE,C 24REQ.

24REQ  IS NOT CATALOGUED OR ASSIGNED
FAC STATUS: 400010000000

@DELETE,C 24TDA.

@ASG,UP   24UIN.,F40,8440B . REQUIREMENTS DELETED BASED ON UIN
```

```
@ASG,UP   24REQ.,F40,8440B  . REQUIREMENTS TO BE MATCHED

@ASG,UP   24TDA.,F40,8440C  . TDA UNITS DELETED FROM REQUIREMENTS

@USE   11,24UIN.

@USE   12,24REQ.

@USE   13,24TDA.

@ASG,A    .F5RS.

@USE TPF$,.F5RS.

@XQT
```

FIRST TIME PEROID = 1

*** HAND-PLAYED-UIN-TABLE ***

SEQ	UIN	TP
1	4	0

RSORT/TEST *** UNCLASSIFIED *** DATE 081976 PAGE 4

SEQ	UIC	TP	COUNT
1	4	0	1

```
••• T O T A L S •••

HAND-PLAYED-UIN   =    1
TDA-UNITS         =    1
REQUIREMENTS      =   18

TOTAL             =   20


@HDG        RSORT UIN FILE     ••• UNCLASSIFIED •••


@DATA,L     11.
DATA T7 RL70-5 08/19-10:24:12
    1.        17052 00   5 H0 0 0 1 HHC  ARMD  CAV RG T W/AT 1000 2 2 2 00000 00030 00020 00188 00238    4 AR 238
END DATA.


@HDG        RSORT REQ FILE     ••• UNCLASSIFIED •••


@DATA,L     12.
DATA T7 RL70-5 08/19-10:24:12
    1.        5035 99   2 H5 0 0 1 BN  COMBAT                  0010 2 1 2 31300 00040 00000 00745 00785 270 EN 785     100
    2.        5035 99   2 H5 0 0 1 BN  COMBAT                  0010 2 1 2 31300 00040 00000 00745 00785 270 EN 785     200
    3.        5035 99   3 H5 0 0 1 BN  COMBAT                  0010 2 1 2 31300 00040 00000 00745 00785 270 EN 785     300
    4.        5115 99   2 H5 0 0 1 BN  CMBT H VY               0011 5 1 2 31300 00043 00000 00804 00847 290 EN 847     400
    5.        5115 99   2 H5 0 0 1 BN  CMBT H VY               0011 5 1 2 31300 00043 00000 00804 00847 290 EN 847     500
    6.        5115 99   3 H5 0 0 1 BN  CMBT H VY               0011 5 1 2 31300 00043 00000 00804 00847 290 EN 847     600
    7.        5115 99   3 H5 0 0 1 BN  CMBT H VY               0011 5 1 2 31300 00043 00000 00804 00847 290 EN 847     700
    8.        5115 99   4 H5 0 0 1 BN  CMBT H VY               0011 5 1 2 31300 00043 00000 00804 00847 290 EN 847     800
    9.        5115 99   4 H5 0 0 1 BN  CMBT H VY               0011 5 1 2 31300 00043 00000 00804 00847 290 EN 847     900
   10.        5327 99   4 G6 0 0 1 CO  TOPOGR APHIC   CORPS    0010 2 1 2 31531 00004 00004 00138 00146 308 EN 147    1000
   11.        5510 FA   2 H2 0 1 1 TM   FIREF IGHTIN G HQ      0010 5 1 2 31572 00001 00000 00003 00004 324 EN   4    11FA
   12.        5510 FA   2 H2 0 1 1 TM   FIREF IGHTIN G HQ      0010 5 1 2 31572 00001 00000 00003 00004 324 EN   4    12FA
   13.        5510 FB   2 H2 0 1 1 TM   FIRE  TRUCK            0010 3 1 2 31300 00000 00000 00006 00006 322 EN   6    13FB
   14.        5510 FB   2 H2 0 1 1 TM   FIRE  TRUCK            0010 5 1 2 31300 00000 00000 00006 00006 324 EN   6    14FB
   15.        5510 FB   2 H2 0 1 1 TM   FIRE  TRUCK            0010 5 1 2 31300 00000 00000 00006 00006 324 EN   6    15FB
   16.        5510 FB   3 H2 0 1 1 TM   FIRE  TRUCK            0010 5 1 2 31300 00000 00000 00006 00006 324 EN   6    16FB
   17.        5510 FB   3 H2 0 1 1 TM   FIRE  TRUCK            0010 5 1 2 31300 00000 00000 00006 00006 324 EN   6    17FB
   18.        5510 FB   3 H2 0 1 1 TM   FIRE  TRUCK            0010 5 1 2 31300 00000 00000 00006 00006 324 EN   6    18FB
END DATA.


@HDG        RSORT TDA FILE     ••• UNCLASSIFIED •••


@DATA,L     13.
DATA T7 RL70-5 08/19-10:24:13
    1.           0      1         1 HHC  THEAT ER ARM Y        0100 5 2 2 39900 00045 00000 00839 00884    2 HQ 884
END DATA.
```

```
@PRT,I
FURPUR 27R1 RL71-3 08/19/76 10:24:13
UNCLASSIFIED*TPF$(0).F40/30,T
UNCLASSIFIED*24TESTRQM(1).F40/30,A  14,
UNCLASSIFIED*15(0).F40/30,T
UNCLASSIFIED*ED$TCB(24E(0).F40/30,T  ED$TC,
UNCLASSIFIED*24UIN(1).F40/30,C  11,
UNCLASSIFIED*24REQ(1).F40/30,C  12,
UNCLASSIFIED*24TDA(1).F40/30,C  13,
UNCLASSIFIED*FSRS(1).F40/30,A  TPF$,
UNCLASSIFIED*XA(0).F40/30,T  $A,
UNCLASSIFIED*XB(0).F14/24,T  $B,


@START  START*24RTEST.FSORT/TEST
```

VIII-8

```
@RUN,/PRNT B124A,H3883P3043K,UNCLASSIFIED,100,500   . FSORT/TEST

@HDG          FSORT/TEST      ***  UNCLASSIFIED ***

@MSG,N  INPUT FILES

@ASG,A      *32TESTFORCE.         . FAS FILE OUTPUT BY SORTUDS

@USE      2,*32TESTFORCE.

@ASG,T   14,F40     . HAND PLAYED UNIT DATA

@ASG,T   15,F40     . LOCKED OUT UNIT DATA

@ASG,T   16,F40      . ALTERNATE THEATER UNIT DATA

@ASG,T   17,F40       . M-CODE 1 (POMCUS) UNIT DATA

@ASG,T   18,F40       . M-CODE 0 (IN-COUNTRY) UNIT DATA

@ASG,T   19,F40        . LOCK OUT COMPO DATA

@USE   TPF$,24TESTDATA.

@MSG,N    LOAD DATA INTO FILES

@ED   HPUIC,14.
CASE UPPER ASSUMED
ED 15.00-08/19/76-10:24:42-(1,)
EDIT
LINES:1 FIELDATA
```

```
@ED   LOAD1,15.
CASE UPPER ASSUMED
ED 15.00-08/19/76-10:24:46-(0,)
EDIT
LINES:1 FIELDATA


@ED   ALTHR,16.
CASE UPPER ASSUMED
ED 15.00-08/19/76-10:24:52-(0,)
EDIT
LINES:1 FIELDATA


@ED   RCMC1,17.
CASE UPPER ASSUMED
ED 15.00-08/19/76-10:24:55-(0,)
EDIT
LINES:1 FIELDATA


@ED   ICMCO,18.
CASE UPPER ASSUMED
ED 15.00-08/19/76-10:24:58-(0,)
EDIT
LINES:2 FIELDATA


@ED   LOCMP,19.
CASE UPPER ASSUMED
ED 15.00-08/19/76-10:25:01-(0,)
EDIT
LINES:2 FIELDATA


@MSG,N  OUTPUT FILES


@DELETE,C   24LOU.
FURPUR 27R1 RL71-3 08/19/76 10:25:03


@DELETE,C   24ATL.


@DELETE,C   24ALT.
```

```
@DELETE,C   24HPU.

@DELETE,C   24CAN.

@ASG,UP     24LOU.,F40,8440C   . LOCKED OUT UNIT FILE

@ASG,UP     24ATL.,F40,8440B          . ABOVE-THE-LINE UNIT FILE

@ASG,UP     24ALT.,F40,8440B           . ALTERNATE THEATER UNIT FILE

@ASG,UP     24HPU.,F40,8440B           . HAND PLAYED UNIT FILE

@ASG,UP     24CAN.,F40,8440C           .MATCH CANDIDATE FILE

@USE     31,24LOU.

@USE     32,24ATL.

@USE     33,24ALT.

@USE     34,24HPU.

@USE     13,24CAN.

@FREE,A TPF$

@QUAL UNCLASSIFIED

@ASG,A      *F5FS.
```


FSORT/TEST ••• UNCLASSIFIED ••• DATE 081976 PAGE 4

```
@USE   TPF$,*FSFS.

@XQT
```

FSORT/TEST ••• UNCLASSIFIED ••• DATE 081976 PAGE 5

••• HAND-PLAYED-UNITS-TABLE •••

SEQ	UIC	ROBCO	ADC01	ADC02	ADC03
1	TEST24		1		059

FSORT/TEST ••• UNCLASSIFIED ••• DATE 081976 PAGE 6

*** LOCKOUT TABLE ***

SEQ	ADC01
1	2

FSORT/TEST   *** UNCLASSIFIED ***   DATE 081976   PAGE 7

••• ALTERNATE-THEATER-TABLE •••

SEQ	ADC01
1	4

*** HCODE = I TABLE ***

SEQ	ROBCO	RDD/ATL	RDD/BTL
1	ROB		075

*** MCODE = 0 TABLE ***

SEQ	COMPO
1	000
2	7

••• HAND-PLAYED-COMPOS-TABLE •••

SEQ	COMPO
1	4
2	5

FSORT/TEST ••• UNCLASSIFIED ••• DATE 081976 PAGE 11

```
••• T O T A L S •••

LOCKOUT              =    1
ABOVE-THE-LINE       =    5
ALT-THEATER          =    2
HAND-PLAYED-UNITS    =    2
MATCH-CANDIDATES     =   13

TOTAL                =   23


@HDG.P   L O C K O U T        •••• UNCLASSIFIED •••


@DATA.L          31.
DATA T7 RL70-5 08/19-10:25:41
    1.      55128002 T10040TESTU1H5200TCCO MEDIUM BOAT        3569IXXXTACOMA   6WA00174206130     1ADC01
END DATA.


@HDG.P   A B O V E  -  T H E  -  L I N E  ••• UNCLASSIFIED •••


@DATA.L          32.
DATA T7 RL70-5 08/19-10:25:42
    1.      17052001 3T10000TEST05H0100ARHHT RGT CAVALRY      18003XXXFT BLISS 5TX00241102018     1NO   NO
    2.      17055101 3T10010TEST06H0100ARSQ CAVALRY           18003XXXFT BLISS 5TX00936102018     2NO   NO
    3.      17055101 3T10010TEST07H0100ARSQ CAVALRY           18003XXXFT BLISS 5TX00936102018     3NO   NO
    4.      17055101 3T10010TEST08H0100ARSQ CAVALRY           18003XXXFT BLISS 5TX00936102018     4NO   NO
    5.      17387001 3T10020TEST09H5100ARCO ATK HEL           18003XXXFT HOOD  5TX00359100000     5NO   NO
END DATA.


@HDG.P   A L T E R N A T E  -  T H E A T E R  •••  UNCLASSIFIED •••


@DATA.L          33.
DATA T7 RL70-5 08/19-10:25:43
    1.      01127001 T10000TEST03H1100AVCO CORPS              30623XXXFT BRAGG 1NC00110400999     1
    2.      01207001 T10000TEST04H2100AVU AIR TRAFFIC CON     30612XXXFT BRAGG 1NC00339403999     2
END DATA.


@HDG.P   H A N D  -  P L A Y E D  •••• UNCLASSIFIED •••


@DATA.L          34.
DATA T7 RL70-5 08/19-10:25:43
    1.      05079001 1T10040TEST20H4100ENCO ASLT FLT BRG RIB  21435R0BFT BLVOIR1VA00176103075     1
```

```
    2.        05124002411008OTEST24G6B00ENCO DUMP TRUCK        31436XXXOROFINO  61D00035104059     2
END DATA.


@HDG.P M A T C H  -  C A N D I D A T E  •••  UNCLASSIFIED  •••


@DATA.L           13.
DATA 17 RL70-5 08/19-10:25:43
    1.        05035991331002OTEST10H5100ENBN COMBAT ARMY       21353XXXFT L WOODSMO00808102015    100
    2.        05035991331003OTEST11H5100ENBN COMBAT ARMY       21353XXXFT CAMPBLIKY00808102021    200
    3.        05035992431005OTEST13H5100ENBN COMBAT ARMY       21353XXXHURON    6SD00785103036    300
    4.        05035992431005OTEST12H5100ENBN COMBAT ARMY       21353XXXASHLAND  5KY00785103046    400
    5.        05035992431006OTEST14H5100ENBN COMBAT ARMY       21353XXXAMERCN FK6UT00785104071    500
    6.        05115991331003OTEST15H3100ENBN COMBAT [HVY]      21364XXXFT RILEY 6KS00845102015    600
    7.        05115992431006OTEST16H3200ENBN COMBAT [HVY]      21364XXXGULFPORT 1MS00847102018    700
    8.        05115992431007OTEST17H3200ENBN COMBAT [HVY]      21364XXXPORTLAND 1ME00847103051    800
    9.        05115992431007OTEST18H3200ENBN COMBAT [HVY]      21364XXXBINGHAMTN1NY00847103026    900
   10.        05115992431008OTEST19H3200ENBN COMBAT [HVY]      21364XXXBAT ROUGESLA00847103041   1000
   11.        05327993431009OTEST21G6200ENCO TOPO CORPS        31531XXXFT BLVOIR1VA00147105101   1100
   12.        05327993431010OTEST22G6200ENCO TOPO CORPS        31531XXXBELL     6CA00147103026   1200
   13.        05510FA3431010OTEST23H2100ENPLT FIREFIGHTING     31557XXXDANVERS  1MA00028104076   13FA
END DATA.
```

VIII-20

```
@PRT.I
FURPUR 27R1 RL71-3 08/19/76 10:25:44
UNCLASSIFIED*TPF$(0),F40/30,T
UNCLASSIFIED*32TESTFORCE(1),F40/30,A   2,
UNCLASSIFIED*14(0),F40/30,T
UNCLASSIFIED*15(0),F40/30,T
UNCLASSIFIED*16(0),F40/30,T
UNCLASSIFIED*17(0),F40/30,T
UNCLASSIFIED*18(0),F40/30,T
UNCLASSIFIED*19(0),F40/30,T
UNCLASSIFIED*ED$TCB124A(0),F40/30,T   ED$TC,
UNCLASSIFIED*24LOU(1),F40/30,C   31,
UNCLASSIFIED*24ATL(1),F40/30,C   32,
UNCLASSIFIED*24ALT(1),F40/30,C   33,
UNCLASSIFIED*24HPU(1),F40/30,C   34,
UNCLASSIFIED*24CAN(1),F40/30,C   13,
UNCLASSIFIED*F5FS(1),F40/30,A   TPF$,
UNCLASSIFIED*XA(0),F40/30,T   $A,
UNCLASSIFIED*XB(0),F14/24,T   $B,


@START  START*24RTEST*ATL/TEST
```

```
@RUN,/PRNT B124B,H3883P6141,UNCLASSIFIED,30,100

@HDG          ATL/TEST                        ••• UNCLASSIFIED •••

@MSG,N  INPUT FILES

@ASG,T 11

@ASG,T 12

@ED  24TESTDATA.SPRDD,11.
CASE UPPER ASSUMED
ED 15•00-08/19/76-11:25:07-(2,)
EDIT
LINES:10 FIELDATA

@ED  24TESTDATA.ATLRDD,12.
CASE UPPER ASSUMED
ED 15•00-08/19/76-11:25:09-(1,)
EDIT
LINES:1 FIELDATA

@ASG,A     24ATL.    . ABOVE-THE-LINE FILE FROM FSORT

@USE    13,24ATL.

@MSG,N  OUTPUT FILES

@DELETE,C 24MATL.
FURPUR 27R1 RL71-3 08/19/76 11:25:10

@ASG,UP    24MATL.,F40,8440B   . MODIFIED ABOVE-THE-LINE FILE (W/RDD)
```

```
@USE   14,24MATL.

@QUAL UNCLASSIFIED

@ASG.A    .F5ATL.

@USE TPF$,.F5ATL.

@XQT
   1      0     0     0     0     0     0     0
   2      2     5     8    11    14    17    17
   3     20    23    26    29    32    35    35
   4     38    41    44    47    50    53    53
   5     56    45    62    65    68    71    71
   6     74    77    80    83    86    89    89
   7    999   999   999   999   999   999   999
   8    999   999   999   999   999   999   999
   9    999   999   999   999   999   999   999
  10    999   999   999   999   999   999   999
   1  18003     1   059   05
 ••• ABOVE-THE-LINE  FINI •••

@HDG           ATL RESULTS             ••• UNCLASSIFIED •••

@DATA,L      14.
DATA T7 RL70-5 08/19-11:25:16
     1.           17052001 3T10000TEST05H0100ARHHT RGT CAVALRY      18003XXXFT BLISS 5TX00241105059    1     NO
     2.           17055101 3T10010TEST06H0100ARSQ CAVALRY           18003XXXFT BLISS 5TX00936105059    2     NO
     3.           17055101 3T10010TEST07H0100ARSQ CAVALRY           18003XXXFT BLISS 5TX00936105059    3     NO
     4.           17055101 3T10010TEST08H0100ARSQ CAVALRY           18003XXXFT BLISS 5TX00936105059    4     NO
     5.           17387001 3T10020TEST09H5100ARCO ATK HEL           18003XXXFT HOOD  5TX00359105059    5     NO
END DATA.

@PRT,I
FURPUR 27R1 RL71-3 08/19/76 11:25:17
UNCLASSIFIED*TPF$(0),F40/30,T
UNCLASSIFIED*11(0),F40/30,T
UNCLASSIFIED*12(0),F40/30,T
UNCLASSIFIED*ED$TCB124B(0),F40/30,T  ED$TC,
UNCLASSIFIED*24ATL(1),F40/30,A  13,
UNCLASSIFIED*24MATL(1),F40/30,C  14,
UNCLASSIFIED*F5ATL(1),F40/30,A  TPF$,
```

```
@RUN,/PRNT B124C,H3883P6141,UNCLASSIFIED,30,100    •ALT/TEST

@HDG          ALT/TEST                       ••• UNCLASSIFIED •••

@MSG,N  INPUT FILES

@ASG,T 11

@ED 24TESTDATA•ALTHR•11•      • ALTERNATE THEATER DATA
CASE UPPER ASSUMED
ED 15•00-08/19/76-10:26:34-(0,)
EDIT
LINES:1 FIELDATA

@ASG,A    24ALT•

@USE   12,24ALT•

@MSG,N  OUTPUT FILES

@DELETE,C 24MALT•
FURPUR 27R1 RL71-3 08/19/76 10:26:36

@ASG,UP   24MALT••F40,8440B    • ALTERNATE THEATER FILE MODIFIED (W/RUD)

@USE   13,24MALT•

@QUAL UNCLASSIFIED

@ASG,A    •F5ALT•
```

```
@USE TPF$,*F5ALT.

@XQT
```


ALT/TEST *** UNCLASSIFIED *** DATE 081976 PAGE 3

```
ADCO1  LOWER-LIMIT  UPPER-LIMIT  INCRE  NEW-ADCO1

  4        010          020        002       4
••• ALTERNATE-THEATER FINI •••


@HDG         ALT RESULTS                 ••• UNCLASSIFIED •••


@DATA,L      13.
DATA T7 RL70-5 08/19-10:26:40
    1.       01127001 T10000TEST03H1100AVCO CORPS             30623XXXFT BRAGG INC00110400010      1
    2.       01207001 T10000TEST04H2100AVU AIR TRAFFIC CON    30612XXXFT BRAGG INC00339400012      2
END DATA.


@PRT,1
FURPUR 27R1 RL71-3 08/19/76 10:26:40
UNCLASSIFIED*TPF$(0),F40/30,T
UNCLASSIFIED*11(0),F40/30,T
UNCLASSIFIED*ED$TC8124C(0),F40/30,T   ED$TC,
UNCLASSIFIED*24ALT(1),F40/30,A   12,
UNCLASSIFIED*24MALT(1),F40/30,C   13,
UNCLASSIFIED*F5ALT(1),F40/30,A   TPF$,


@START  START*24RTEST.MATCH/TEST
```

```
@RUN      B124F,H3883P5141,UNCLASSIFIED,100,500 . MATCH/TEST

@HDG         MATCH/TEST        *** UNCLASSIFIED ***

@MSG,N  INPUT FILES

@ASG,A       24REQ.     . REQUIREMENTS FILE FROM RSORT

@ASG,A       24CAN.      . MATCH CANDIDATES FILE FROM FSORT

@USE       1,24CAN.

@USE       2,24REQ.

@ASG,T     11,F40     . PECOD DATA

@ASG,T     12,F40      . SUBPERIOD RDD DATA

@ASG,T     13,F40     . NOTIONAL UNIT DATA

@USE     TPF$,24TESTDATA.

@ED     PECOD,11.
CASE UPPER ASSUMED
ED 15.00-08/19/76-10:27:09-(0,)
EDIT
LINES:23 FIELDATA

@ED     SPRDD,12.
CASE UPPER ASSUMED
ED 15.00-08/19/76-10:27:10-(1,)
EDIT
LINES:10 FIELDATA
```

```
@ED     NOTIN.13.
CASE UPPER ASSUMED
ED 15.00-08/19/76-10:27:12-(0,)
EDIT
LINES:20 FIELDATA

@FREE,A TPF$

@MSG,N  OUTPUT FILES

@DELETE,C    24MOVR.
FURPUR 27R1 RL71-3 08/19/76 10:27:13

@DELETE,C    24MREQ.

@DELETE,C    24MATCHRPT.

@ASG,UP      24MOVR.,F40,8440A     . THIS FILE NOT USED

@ASG,UP      24MREQ.,F14,PACK61     . MATCH REQUIREMENTS (MATCHES AND NOTIONALS)

@ASG,UP      24MATCHRPT.,F40      . MATCH REPORT FILE

@USE      3,24MOVR.

@USE      8,24MREQ.

@USE      19,24MATCHRPT.

@QUAL UNCLASSIFIED
```

```
@ASG,A      *F5FM.

@USE   TPFS,*F5FM.

@DELETE,C       F5TPRT.
FURPUR 27R1 RL71-3 08/19/76 10:28:35

@ASG,UP         F5TPRT.,F40    . INTERMEDIATE PRINT FILE

@BRKPT  PRINTS/F5TPRT
```

REQ-SRC	FORCE-SRC	BRNCH	LEVEL	UNTDS	COMPO	UIC	TPSN	UIN	TP	RSEQ	MC	ROBCO	STNNM	LOCCO	RSTR	FSTR	FSEQ
05035 00	05035 00	EN	BN	COMBAT ARMY	1	TEST10	21353	270	2	1	3	XXX	FT L WOOD	5MO	00785	00808	1
05035 00	05035 00	EN	BN	COMBAT ARMY	1	TEST11	21353	270	2	2	3	XXX	FT CAMPBL	1KY	00785	00808	2
05035 00	05035 00	EN	BN	COMBAT ARMY	2	TEST13	21353	270	3	3	4	XXX	HURON	6SD	00785	00785	3
	05035 00	EN	BN	COMBAT ARMY	2	TEST12	21353				4	XXX	ASHLAND	5KY		00785	4
	05035 00	EN	BN	COMBAT ARMY	2	TEST14	21353				4	XXX	AMERCN FK	6UT		00785	5
05115 00	05115 00	EN	BN	COMBAT [HVY]	1	TEST15	21364	290	2	4	3	XXX	FT RILEY	6KS	00847	00845	6
05115 00	05115 00	EN	BN	COMBAT [HVY]	2	TEST16	21364	290	2	5	4	XXX	GULFPORT	1MS	00847	00847	7
05115 00	05115 00	EN	BN	COMBAT [HVY]	2	TEST17	21364	290	3	6	4	XXX	PORTLAND	1ME	00847	00847	8
05115 00	05115 00	EN	BN	COMBAT [HVY]	2	TEST18	21364	290	3	7	4	XXX	BINGHAMTN	1NY	00847	00847	9
05115 00	05115 00	EN	BN	COMBAT [HVY]	2	TEST19	21364	290	4	8	4	XXX	BAT ROUGE	5LA	00847	00847	10
05115 00		EN	BN	CMBT HVY	4	X50001	31300	290	4	9			FT RILEY	6KS	00847	00847	
05327 00	05327 00	EN	CO	TOPO CORPS	3	TEST21	31531	308	4	10	4	XXX	FT BLVOIR	1VA	00147	00147	11
	05327 00	EN	CO	TOPO CORPS	3	TEST22	31531				4	XXX	BELL	6CA		00147	12
05510 FA	05510 FA	EN	PLT	FIREFIGHTING	3	TEST23	31557	324	2	11	4	XXX	DANVERS	1MA	00004	00028	13
05510 FA		EN	TM	FIREFIGHTING HQ	4	X50002	31572	324	2	12			FT RILEY	6KS	00004	00004	
05510 FB		EN	TM	FIRE TRUCK	4	X50003	31300	322	2	13			FT RILEY	6KS	00006	00006	
05510 FB		EN	TM	FIRE TRUCK	4	X50004	31300	324	2	14			FT RILEY	6KS	00006	00006	
05510 FB		EN	TM	FIRE TRUCK	4	X50005	31300	324	2	15			FT RILEY	6KS	00006	00006	
05510 FB		EN	TM	FIRE TRUCK	4	X50006	31300	324	3	16			FT RILEY	6KS	00006	00006	
05510 FB		EN	TM	FIRE TRUCK	4	X50007	31300	324	3	17			FT RILEY	6KS	00006	00006	
05510 FB		EN	TM	FIRE TRUCK	4	X50008	31300	324	3	18			FT RILEY	6KS	00006	00006	

```
@RUN /PRNT B124G,H3883P6141,UNCLASSIFIED,100,300 . LAYIN/TEST

@HDG        LAYIN                   *** UNCLASSIFIED ***

@MSG,N  INPUT FILES

@ASG,T        10.

@ED         24TESTDATA.SPRDD,10.
CASE UPPER ASSUMED
ED 15.00-08/19/76-11:26:38-(2,)
EDIT
LINES:10 FIELDATA

@ASG,A      24MALT.   . ALTERNATE THEATER UNITS FROM ALT

@ASG,A      24MATL.   . ABOVE-THE-LINE UNITS FROM ATL

@ASG,A      24LOU.   . LOCK OUT UNITS FROM FSORT

@ASG,A      24HPU.   . HAND PLAYED UNITS FROM FSORT

@ASG,A      24MREQ.   . MATCHES AND NOTIONALS FROM MATCH

@ASG,A      32TESTFORCE.   . FORCE FILE FROM SORTUDS

@USE     11,24MALT.

@USE     12,24MATL.

@USE     13,24LOU.
```

```
@USE     14,24HPU.

@USE     15,24MREQ.

@USE     16,32TESTFORCE.

@MSG,N  OUTPUT FILES

@DELETE,C    24FOUT.
FURPUR 27R1 RL71-3 08/19/76 11:28:12

@DELETE,C    24FPRT.

@ASG,UP      24FOUT.,F14,PACK62   . FORCE OVERLAID WITH MATCH RESULTS

@MSG,N      24FOUT WILL BE INPUT TO MRG

@ASG,UP      24FPRT.,F40  . INTERMEDIATE PRINT FILE

@USE     17,24FOUT.

@USE     18,24FPRT.

@QUAL UNCLASSIFIED

@ASG,A       .F5LAY

@USE    TPF$,.F5LAY
```

```
@XQT
    1     0     0     0     0     0     0     0
    2     2     5     8    11    14    17    17
    3    20    23    26    29    32    35    35
    4    38    41    44    47    50    53    53
    5    56    45    62    65    68    71    71
    6    74    77    80    83    86    89    89
    7   999   999   999   999   999   999   999
    8   999   999   999   999   999   999   999
    9   999   999   999   999   999   999   999
   10   999   999   999   999   999   999   999
  TALT=     2 TATLOV=     0 TATL=     5 TBYPAS=     0 TLKOUT=     1

  THP=      2 TSHORT=     8 TMATCH=   10 TOVER=     3 TOTAL =     31

  *** LAYIN FINI ***

@FREE      24FPRT.

@MSG,N @SYM 24FPRT.

@END
@END IGNORED - IN CONTROL MODE

@PRT,I
FURPUR 27R1 RL71-3 08/19/76 11:31:25
UNCLASSIFIED*TPF$(0),F40/30,T
UNCLASSIFIED*10(0),F40/30,T
UNCLASSIFIED*ED$TCB124G(0),F40/30,T  ED$TC,
UNCLASSIFIED*24MALT(1),F40/30,A  11,
UNCLASSIFIED*24MATL(1),F40/30,A  12,
UNCLASSIFIED*24LOU(1),F40/30,A  13,
UNCLASSIFIED*24HPU(1),F40/30,A  14,
UNCLASSIFIED*24MREQ(1),F14/24,A  15,
UNCLASSIFIED*32TESTFORCE(1),F40/30,A  16,
UNCLASSIFIED*24FOUT(1),F14/24,C  17,
UNCLASSIFIED*F5LAY(1),F40/30,A  TPF$,
UNCLASSIFIED*XA(0),F40/30,T  $A,
UNCLASSIFIED*XB(0),F14/24,T  $B,

@MSG,N START*24R*BINCOPY-SUM
```

VIII-32

```
@RUN,/R C124A,H3883P5121,UNCLASSIFIED,15,500  . GEORUN

@HDG   GEO RUN

@ASG,A  24FOUT.     . FORCE FILE OUTPUT BY FMA PROGRAM LAYIN

@USE 17,24FOUT.

@QUAL  UNCLASSIFIED

@ASG,A •32GEO.     . GEO FILE OUTPUT BY SORTARLOC

@ASG,A •32NODE.      . NODE TABLE

@ASG,T  32F7.              . PRINT OF MDF1 FILE

@ASG,A  32F29              . MDF1 FILE - OUTPUT (SRC SEQUENCE)

@USE 15,•32GEO.

@USE 16,•32NODE.

@USE 7,32F7.

@USE 29,32F29.

@ASG,XA •58CAMP.

@PRT,S  •24TESTDATA.GEODATA
```

FURPUR 27R1 RL71-3 08/25/76 08:58:24

GEO RUN DATE 082576 PAGE 3

```
UNCLASSIFIED*24TESTDATA(1).GEODATA
     1       2
     2       1 000 XPQK 004 000
     3       4 010 WTGR 012 010
     4       1
     5       18003                  A    HCRL


@XQT     *58CAMP.GEOABS


@ADD.P  *24TESTDATA.GEODATA
 NUMBER OF THEATERS     2
 NUMBER OF EXCEPTIONS   1

 THTR AVAIL DEST(G) DEST(N) AVL(T)
   1     0   XPQK       4     0
   4    10   WTGR      12    10

 TPSN  DEST(G)  DEST(N)  AVL  RDD  MODE  THR  ORIG(G)  ORIG(N)
 18003               0     0    0    A     0    HCRL       0
    8800 RECORDS ON FILE BIGEO
      83 RECORDS ON FILE BINODE
        UNITS       ACTIVE      NO FIND      RESERVE      NO FIND   AVE ACC/U
          27          11           0           16            0       15.852
           0


@PRT.I
FURPUR 27R1 RL71-3 08/25/76 09:00:23
UNCLASSIFIED*TPFS(0).F40/30,T
UNCLASSIFIED*24FOUT(1).F14/24,A  17,
UNCLASSIFIED*32GEO(1).F14/24,A  15,
UNCLASSIFIED*32NODE(1).F14/24,A  16,
UNCLASSIFIED*32F7(0).F40/30,T  7,
UNCLASSIFIED*32F29(1).F40/30,A  29,
UNCLASSIFIED*58CAMP(1).F14/24,AX  CSINTNAMES,
UNCLASSIFIED*XA(0).F40/30,T  $A,
UNCLASSIFIED*XB(0).F14/24,T  $B,
UNCLASSIFIED*21(0).F40/30,T


@ED.R  32F7.
READ-ONLY MODE
CASE UPPER ASSUMED
ED 15.00-08/25/76-09:00:24-(0,)
EDIT
 01127H10000  TEST03 30623  HCTL37 WTGR  190  12  10  10 P 4 3 CORPS          @
 01207H20000  TEST04 30612  HCTL37 WTGR  190  12  12  10 P 4 3 IR TRAFFIC C   @
 17052H00000  TEST05 18003  HCRL   XPQK  210   4  59   0 A 1 3  RGT CAVALRY   @
 17055H01000  TEST06 18003  HCRL   XPQK  210   4  59   0 A 1 3 CAVALRY        @
 17055H01000  TEST07 18003  HCRL   XPQK  210   4  59   0 A 1 3 CAVALRY        @
 17055H01000  TEST08 18003  HCRL   XPQK  210   4  59   0 A 1 3 CAVALRY        @
```

```
17387H50000  TEST09 18003    HCRL    XPQK   210   4   59   0 A 1 3 ATK HEL        @
05035H50000  TEST10 21353    HGSH29  XPQK   186   4    2   0 P 1 3 COMBAT ARMY    @
05035H50000  TEST11 21353    HDBL21  XPQK   227   4    2   0 P 1 3 COMBAT ARMY    @
05115H30000  TEST15 21364    HKBN20  XPQK   178   4    5   0 P 1 3 COMBAT [HVY]   @
05079H40000  TEST20 21435 E  HCHLS1  XPQK   192   4   75   0 A 1 1 ASLT FLT BRG   @
05035H50000  TEST13 21353    HFTZ48  XPQK   210   4   20   0 P 1 3 COMBAT ARMY    @
05115H30000  TEST16 21364    HFTZ48  XPQK   210   4    5   0 P 1 3 COMBAT [HVY]   @
05115H30000  TEST17 21364    HFTZ48  XPQK   210   4   23   0 P 1 3 COMBAT [HVY]   @
05115H30000  TEST18 21364    HFTZ48  XPQK   210   4   26   0 P 1 3 COMBAT [HVY]   @
05115H30000  TEST19 21364    HFTZ48  XPQK   210   4   38   0 P 1 3 COMBAT [HVY]   @
05124G60000  TEST24 31436    HFTZ48  XPQK   210   4   59   0 P 1 3 DUMP TRUCK     @
05327G60000  TEST21 31531    HFTZ48  XPQK   210   4   41   0 P 1 3 TOPO CORPS     @
05510H2FA00  TEST23 31557    HFTZ48  XPQK   210   4    8   0 P 1 3  FIREFIGHTIN   @
05115H50000  X50001 31300    HKBN20  XPQK   178   4   44   0 P 1 3 CMBT HVY       @
05510H2FA01  X50002 31572    HKBN20  XPQK   178   4    8   0 P 1 3  FIREFIGHTIN   @
05510H2FB01  X50003 31300    HKBN20  XPQK   178   4   11   0 P 1 3  FIRE TRUCK    @
05510H2FB01  X50004 31300    HKBN20  XPQK   178   4   14   0 P 1 3  FIRE TRUCK    @
05510H2FB01  X50005 31300    HKBN20  XPQK   178   4   17   0 P 1 3  FIRE TRUCK    @
05510H2FB01  X50006 31300    HKBN20  XPQK   178   4   29   0 P 1 3  FIRE TRUCK    @
05510H2FB01  X50007 31300    HKBN20  XPQK   178   4   32   0 P 1 3  FIRE TRUCK    @
05510H2FB01  X50008 31300    HKBN20  XPQK   178   4   35   0 P 1 3  FIRE TRUCK    @
EOF:27 SCAN:26
NO CORRECTIONS APPLIED.


@FREE  32F7.


@FREE  32F29.


@MSG,N START*24RTEST.LOGSUBRUN
```

```
@RUN,/PRNT A124B,H3883P6141,UNCLASSIFIED,30,30 . LOGSUBRUN

@HDG     UPDATING MDF1 SRC'S   -  LOG SUB RUN

@QUAL UNCLASSIFIED

@ASG,AX *32CAMP.

@ASG,A    32F29.     . MDF1 FILE WITH BAD MATCH ON TUCHA

@USE 29.,32F29.

@ASG,A    32F29UP.   . NEW MDF1 WITH FIXED UP SRC'S

@USE 25.,32F29UP.

@PRT,S  *24TESTDATA.LOGSUBDATA
FURPUR 27R1 RL71-3 08/24/76 18:59:19
```

```
UNCLASSIFIED*24TESTDATA(1).LOGSUBDATA
    1        1
    2        05115H50000 05115H30000

@XQT     .32CAMP.UPMDF1

@ADD.P     .24TESTDATA.LOGSUBDATA

@PRT,I
FURPUR 27R1 RL71-3 08/24/76 18:59:25
UNCLASSIFIED*TPFS(0).F40/30,T
UNCLASSIFIED*32CAMP(1).F14/24,AX   CSINTNAMES,
UNCLASSIFIED*32F29(1).F40/30,A   29,
UNCLASSIFIED*32F29UP(1).F40/30.A   25,
UNCLASSIFIED*XA(0).F40/30,T   SA,
UNCLASSIFIED*XB(0).F14/24,T   SB,

@START START*24RTEST.LOGUPRUN
```

```
@RUN,/PRNT A124C,H3883P6141,UNCLASSIFIED,50,50 . LOGUPRUN
@HDG     LOG RUN
@QUAL UNCLASSIFIED
@ASG,XA *58CAMP.
@ASG,A 32F29UP.    . UPDATED MDF1 FILE
@ASG,A *32TUSUM.                  . TUCHA FILE ( SRC SEQUENCE)
@ASG,A 24POMSUM.
@ASG,A 32F26.          . MDF2 FILE - OUTPUT
@ASG,A 32F11.          . INTERMEDIATE FILE FOR RPG
@ASG,A 32F10.          . UNIT MORSA CARDS - OUTPUT
@USE 11.,32F11.
@USE 10.,32F10.
@USE 29.,32F29UP.
@USE 28.,*32TUSUM.
@USE 27.,24POMSUM.
```

```
@USE 26..32F26.

@PRT.S •24TESTDATA.LOGDATA
FURPUR 27R1 RL71-3 08/24/76 18:59:48
```

DATE 082476 PAGE 3

```
UNCLASSIFIED*24TESTDATA(1).LOGDATA
   1        2 2 14        12        12
   2                 200.0 100.0    25.0  25.0
   3        @ADD,P  24TESTDATA.LOGSUBDATA

@XQT *58CAMP.LOGARS

@ADD,P  24TESTDATA.LOGSUBDATA
 NUMBER OF THEATERS  2
 NUMBER OF SETS      2
 THTR  SET  PLANID
   1    1     1
   4    2     2
     AIRPORT      SEAPORT         NON-POMCUS ASR              POMCUS ASR
                               SUPPLY        AMMO      SUPPLY       AMMO
                              200.000       25.000      .000       25.000

 SRC LIST
  05115H50000   05115H30000
   UNITS IN THEATER SETS      24     2      0      0      0      0      0
 FINAL SEQUENCE NUMBERS       24     2      0      0      0      0      0
 NO MATCHES                    0
   UNITS IN THEATER SETS      25     2      0      0      0      0      0
 FINAL SEQUENCE NUMBERS       26     2      0      0      0      0      0
 NO MATCHES                    0

@PRT,I
FURPUR 27R1 RL71-3 08/24/76 19:00:25
UNCLASSIFIED*TPF$(0),F40/30,T
UNCLASSIFIED*58CAMP(1),F14/24,AX  CSINTNAMES,
UNCLASSIFIED*32F29UP(1),F40/30,A  29,
UNCLASSIFIED*32TUSUM(1),F40/30,A  28,
UNCLASSIFIED*24POMSUM(1),F40/30,A  27,
UNCLASSIFIED*32F24(1),F40/30,A  26,
UNCLASSIFIED*32F11(1),F40/30,A  11,
UNCLASSIFIED*32F10(1),F40/30,A  10,
UNCLASSIFIED*XA(0),F40/30,T  SA,

@ED,R 10.
READ-ONLY MODE
CASE UPPER ASSUMED
ED 15.00-08/24/76-19:00:25-(0,)
EDIT
0001A2A     U7CORPS          CO P  010010       HCTLWTGRWTGR000102030623
0001A2W0000100000210001080000280000000000010112 7H10000419 03AAE7TEST03
0001A2FN                 000016002839
0002A2A     U7IR TRAFFIC CU   P  010012       HCTLWTGRWTGR000338030612
0002A2W000034000047000316000003000000000000401207H20000419 03W9ACTEST04
0002A2FN                 000009001629
```

```
0001A1A     U6COMBAT ARMY BN P  000002       HDBLXPQKXPQK000780021353
0001A1W000078000130001616000392000000000001005035H5000012274BAATTEST11
0001A1FN                000003000485
0002A1A     U6COMBAT ARMY BN P  000020       HFTZXPQKXPQK000780021353
0002A1W000078000130001616000392000000000001005035H500001210480AATTEST13
0002A1FN                000003000485
0003A1A     U6COMBAT ARMY BN P  000002       HGSHXPQKXPQK000780021353
0003A1W000078000130001616000392000000000001005035H5000011864BAATTEST10
0003A1FN                000003000485
0004A1A     U6COMBAT CHVYJBN P  000005       HKBNXPQKXPQK000847021364
0004A1W0000850001520016690017720000000000011051I5H30000117A4X7METEST15
0005A1A     U7CMBT HVY   BN P   000044       HKBNXPQKXPQK000847031300
0005A1W0000850001520016690017720000000000011051I5H50000117A4X7MEX50001
0006A1A     U6COMBAT CHVYJBN P  000026       HFTZXPQKXPQK000847021364
0006A1W0000850001520016690017720000000000011051I5H3000012104X7METEST18
0007A1A     U6COMBAT CHVYJBN P  000038       HFTZXPQKXPQK000847021364
0007A1W0000850001520016690017720000000000011051I5H3000012104X7METEST19
0008A1A     U6COMBAT CHVYJBN P  000023       HFTZXPQKXPQK000847021364
0008A1W0000850001520016690017720000000000011051I5H3000012104X7METEST17
0009A1A     U6COMBAT CHVYJBN P  000005       HFTZXPQKXPQK000847021364
0009A1W0000850001520016690017720000000000011051I5H3000012104X7METEST16
0010A1A     U7DUMP TRUCK  CO P  000059       HFTZXPQKXPQK000107031436
0010A1W00001100001300058500002000000000000105124G60000121O4XZAATEST24
0011A1A     U7TOPO CORPS  CO P  000041       HFTZXPQKXPQK000146031531
0011A1W000015000045000169000202000000000002053276600001210 4WOAATEST21
0011A1FN                000002000365
0012A1A     U7 FIREFIGHTINTM P  000008       HFTZXPQKXPQK000004031557
0012A1W0000000000001000004000000000000000000551OH2FA001210 49ZFETEST23
0013A1A     U7 FIREFIGHTINTM P  000008       HKBNXPQKXPQK000004031572
0013A1W0000000000001000004000000000000000000551OH2FA01117A49ZFEX50002
0014A1A     U7 FIRE TRUCK TM P  000032       HKBNXPQKXPQK000006031300
0014A1W000001000000000000900000000000000000551OH2FB01117A49ZFFX50007
0015A1A     U7 FIRE TRUCK TM P  000035       HKBNXPQKXPQK000006031300
0015A1W000001000000000000900000000000000000551OH2FB01117A49ZFFX50008
0016A1A     U7 FIRE TRUCK TM P  000014       HKBNXPQKXPQK000006031300
0016A1W000001000000000000900000000000000000551OH2FB01117A49ZFFX50004
0017A1A     U7 FIRE TRUCK TM P  000017       HKBNXPQKXPQK000006031300
0017A1W000001000000000000900000000000000000551OH2FB01117A49ZFFX50005
0018A1A     U7 FIRE TRUCK TM P  000029       HKBNXPQKXPQK000006031300
0018A1W000001000000000000900000000000000000551OH2FB01117A49ZFFX50006
0019A1A     U7 FIRE TRUCK TM P  000011       HKBNXPQKXPQK000006031300
0019A1W000001000000000000900000000000000000551OH2FB01117A49ZFFX50003
0020A1A     U8 RGT CAVALRYHHTA  000059       HCRLXPQKXPQK00023801A003
0020A1W0000240000520003290000400000000000003170 52H00000121O2UB81TEST05
0020A1FN                000018003160
0021A1A     U8CAVALRY     SQ A  000059       HCRLXPQKXPQK00093801A003
0021A1W000094000095001848001347000000000001217055H0100012102UB86TEST08
0021A1FN                000006000970
0022A1A     U8CAVALRY     SQ A  000059       HCRLXPQKXPQK00093801A003
0022A1W000094000095001848001347000000000001217055H0100012102UB86TEST06
0022A1FN                000006000970
0023A1A     U8CAVALRY     SQ A  000059       HCRLXPQKXPQK00093801A003
0023A1W000094000095001848001347000000000001217055H0100012102UB86TEST07
0023A1FN                000006000970
0024A1A     U8ATK HEL     CO A  000059       HCRLXPQKXPQK00020101A003
0024A1W000020000034000099000017000000000003173 87H5000012102DAAATEST09
0024A1FN                000015002535
```

```
0025A1A     U6ASLT FLT BRGCO A  000075      HCHLXPQKXPQK00009102;435
0025A1W00000000000030000140000000000000000000105079H40000011924BAAGTEST20
0025A1FN              0000;3002178
0025A1FA E
0026A1A     U6ASLT FLT BRGCO Z  999000      HCHLXPQKXPQK00000002;435
0026A1W00000000000090000920000080000000000000005079H40000011924BAAGTEST20
0026A1FA E
EOF:70 SCAN:69
NO CORRECTIONS APPLIED.


@START START•24RTEST.NURRUN
```


LOG RUN DATE 082476 PAGE 6

```
@RUN,/PRNT A124D,H3883P6141,UNCLASSIFIED,100,300 . NURRUN

@HDG   NUR RUN

@QUAL UNCLASSIFIED

@ASG,A 32F26.      . MDF2 FILE (NON-POMCUS IN SRC / POMCUS IN UIC SEQ

@USE 26,32F26

@ASG,A 32F24.      . NUR MORSA CARDS - OUTPUT (RDD SEQUENCE)

@USE 24,32F24

@ASG,A 32F23       . NUR DATA FOR PCKAGE - OUTPUT (RDD SEQUENCE)

@USE 23,32F23

@ASG,A 32F12.     . INTERMEDIATE FILE FOR RPG

@USE 12,32F12

@ASG,XA *58CAMP

@PRT,S *24TESTDATA.NURDATA
FURPUR 27R1 RL71-3 08/24/76 19:00:46
```

```
UNCLASSIFIED*24TESTDATA(1).NURDATA
      1        2 2 14        12        12
      2        1  18  28  75  00  35  17  35  53  71  89  89  89  89  25  30  25  10  10  10
      3        2  18  28  75  00  35  17  35  53  71  89  89  89  89  25  30  25   2   2   2
      4        1    5000.0       800.0        40000.0  0.0                             XPQK   4  10
      5        2    1000.0       200.0         1000.0  0.0                             WTGR  12  10
      6        1 0.90   0.80   0.70   0.60   0.60   0.60   0.60   0.60
      7        2 0.77   0.77   0.77   0.77   0.77   0.77   0.77   0.77
      8        1 200.0    25.0      0.0  100.0    25.0    0.0
      9        2 200.0    25.0      0.0  100.0    25.0    0.0
     10        11 30.00   50.00   40.00   30.00   30.00   30.00   30.00   30.00
     11        12  0.0   100.0    50.0    50.0    20.0    20.0    10.0    10.0
     12        13 45.00   45.00   45.00   45.00   45.00   45.00   45.00   45.00
     13        21 29.00   29.00   29.00   29.00   29.00   25.00   25.00   25.00
     14        22 22.63   22.63   22.63   22.63   22.63   22.00   22.00   22.00
     15        23 55.00   55.00   55.00   45.00   45.00   45.00   45.00   45.00
     16        RPCA 216 1.0   0.0   NEW CUMBERLAND
     17        AKWQ 174  0.0 1.0    ANNISTON - AMMO
     18                  0.0  0.0   NOT USED
     19                  0.0  0.0   NOT USED
     20                  0.0  0.0   NOT USED
     21                  0.0  0.0   NOT USED
     22                  0.0  0.0   NOT USED
     23                  0.0  0.0   NOT USED
     24                  0.0  0.0   NOT USED
     25                  0.0  0.0   NOT USED
     26                  0.0  0.0   NOT USED
     27                  0.0  0.0   NOT USED
     28                  0.0  0.0   NOT USED
     29                  0.0  0.0   NOT USED
     30                  0.0  0.0   NOT USED
     31                  0.0  0.0   NOT USED
     32                  0.0  0.0   NOT USED
     33                  0.0  0.0   NOT USED
     34        HQLH 191  0.0  0.0   GALVESTON - POL
     35        HEKP 187  0.0  0.0   FT DIX - PAX


@XQT *58CAMP.NURABS


@ADD,P *24TESTDATA.NURDATA
 NUMBER OF THEATERS  2
 NUMBER OF SETS      2

 THTR SET
   1   1

   4   2

 SET PLAN DDAY RDAY BLDUP CDAY ADAY  1   2   3   4   5   6   7   8    BLDUP LEVELS     SAFE LEVELS
  1   1    18   28    75    0   35  17  35  53  71  89  89  89  89   25.  30.  25.   10.  10.  10.
  2   2    18   28    75    0   35  17  35  53  71  89  89  89  89   25.  30.  25.    2.   2.   2.

 SET          THEATER BASE              FILLER     FILLER        FILLER                  DAY
```

VIII-45

	DRY	AMMO	POL	PCT	DAYS			PCTS			DEST(G)	DEST(N)	AVL
1	5000.000	800.000	40000.000	.000	0	0	0	.000	.000	.000	XPQK	4	10
2	1000.000	200.000	1000.000	.000	0	0	0	.000	.000	.000	WTGR	12	10

REPLACEMENTS BY TIME PERIOD (MEN/1000/DAY)

1	.90000	.80000	.70000	.68000	.60000	.60000	.60000	.60000
2	.77000	.77000	.77000	.77000	.77000	.77000	.77000	.77000

ACCOMPANYING SUPPLY RATES (LBS/MAN)

	NON-POMCUS			POMCUS		
1	200.00000	25.00000	.00000	100.00000	25.00000	.00000
2	200.00000	25.00000	.00000	100.00000	25.00000	.00000

CONSUMPTION RATES (LBS/MAN/DAY) - BY TIME PERIOD

1	DRY	30.00000	50.00000	40.00000	30.00000	30.00000	30.00000	30.00000	30.00000
	AMMO	.00000	100.00000	50.00000	50.00000	20.00000	20.00000	10.00000	10.00000
	POL	45.00000	45.00000	45.00000	45.00000	45.00000	45.00000	45.00000	45.00000
2	DRY	29.00000	29.00000	29.00000	29.00000	29.00000	25.00000	25.00000	25.00000
	AMMO	22.63000	22.63000	22.63000	22.63000	22.63000	22.00000	22.00000	22.00000
	POL	55.00000	55.00000	55.00000	45.00000	45.00000	45.00000	45.00000	45.00000

ORIG(G)	ORIG(N)	DRY PCT	AMMO PCT
RPCA	216	1.000	.000
AKWQ	174	.000	1.000
	0	.000	.000
	0	.000	.000
	0	.000	.000
	0	.000	.000
	0	.000	.000
	0	.000	.000
	0	.000	.000
	0	.000	.000
	0	.000	.000
	0	.000	.000
	0	.000	.000
	0	.000	.000
	0	.000	.000
	0	.000	.000
	0	.000	.000
	0	.000	.000
HQLH	191	.000	.000
HEKP	187	.000	.000

DAY	THR	UNITS	STRENGTH	REPLACEMNTS	TYPE	DAYS OH	DAYS DESRD	TONS OH	TONS REQRD
1	1	0	0.						
1	2	0	0.						
18	1	9	3280.	0.	2	4.	10.	677.	963.
19	1	9	3280.	3.	2	9.	10.	1476.	164.
20	1	10	4060.	5.	2	7.	10.	1447.	583.
20	2	2	440.	1.	3	2.	2.	24.	0.
21	1	10	4060.	8.	2	9.	10.	1827.	203.
21	2	2	440.	1.	3	1.	2.	12.	0.
22	1	10	4060.	12.	2	9.	10.	1827.	203.
22	2	2	440.	1.	3	1.	2.	12.	0.
23	1	11	4907.	15.	2	7.	10.	1795.	658.
23	2	2	440.	2.	3	1.	2.	12.	0.
24	1	11	4907.	19.	2	9.	10.	2208.	245.
24	2	2	440.	2.	3	1.	2.	12.	0.
25	1	11	4907.	23.	2	9.	10.	2208.	245.
25	2	2	440.	2.	3	1.	2.	12.	0.
26	1	12	5754.	27.	2	8.	10.	2176.	701.
26	2	2	440.	3.	3	1.	2.	12.	0.
27	1	12	5754.	31.	2	9.	10.	2589.	288.
27	2	2	440.	3.	3	1.	2.	12.	0.
30	1	13	5760.	45.	1	24.	25.	3600.	143.
30	1	13	5760.	45.	2	8.	11.	3125.	824.
30	1	13	5760.	45.	3	25.	25.	3240.	0.
30	2	2	440.	4.	2	23.	25.	125.	11.
30	2	2	440.	4.	3	0.	3.	36.	0.
35	1	15	5772.	23.	1	20.	25.	3607.	727.
35	1	15	5772.	23.	2	6.	13.	3746.	2061.
35	1	15	5772.	23.	3	20.	25.	3247.	5.
35	2	2	440.	2.	2	20.	26.	127.	27.
35	2	2	440.	2.	3	-2.	5.	66.	1.
40	1	16	6619.	22.	1	23.	25.	3309.	257.
40	1	16	6619.	22.	3	17.	25.	3723.	8.
40	2	2	440.	2.	2	21.	26.	130.	28.
40	2	2	440.	2.	3	0.	8.	95.	1.
45	1	18	7612.	24.	1	18.	25.	3806.	1088.
45	1	18	7612.	24.	2	10.	17.	3280.	1282.
45	1	18	7612.	24.	3	17.	25.	4282.	10.
45	2	2	440.	2.	2	21.	27.	133.	28.
45	2	2	440.	2.	3	3.	10.	125.	1.
50	1	18	7612.	27.	1	20.	25.	3806.	761.
50	1	18	7612.	27.	2	12.	19.	3685.	1356.
50	1	18	7612.	27.	3	20.	25.	4282.	6.
50	2	2	440.	2.	2	22.	27.	136.	28.
50	2	2	440.	2.	3	5.	13.	154.	1.
55	1	18	7612.	26.	2	14.	21.	4089.	1356.
55	1	18	7612.	26.	3	20.	25.	4282.	6.
55	2	2	440.	2.	2	22.	28.	138.	28.
55	2	2	440.	2.	3	10.	15.	151.	0.
60	1	24	10972.	25.	1	17.	25.	4114.	1317.
60	1	24	10972.	25.	2	11.	24.	6478.	3382.
60	1	24	10972.	25.	3	14.	25.	6172.	20.

NUR RUN DATE 082476 PAGE 5

DAY	THR	UNITS	STRENGTH	REPLACEMNTS	TYPE	DAYS OH	DAYS DESRD	TONS OH	TONS REQRD
60	2	2	440.	2.	2	23.	28.	141.	28.
60	2	2	440.	2.	3	10.	18.	175.	1.
65	1	24	10972.	33.	1	20.	25.	4114.	823.
65	1	24	10972.	33.	2	19.	26.	7062.	1955.
65	1	24	10972.	33.	3	20.	25.	6172.	9.
65	2	2	440.	2.	2	23.	29.	144.	28.
65	2	2	440.	2.	3	13.	20.	199.	1.
70	1	24	10972.	33.	1	20.	25.	4114.	823.
70	1	24	10972.	33.	2	21.	28.	7645.	1955.
70	1	24	10972.	33.	3	20.	25.	6172.	9.
70	2	2	440.	2.	2	24.	29.	147.	28.
70	2	2	440.	2.	3	15.	23.	223.	1.
END OF MDF2 FILE. DAY = 75									
75	1	25	11063.	33.	1	20.	25.	4149.	852.
75	1	25	11063.	33.	3	20.	25.	6223.	9.
75	2	2	440.	2.	2	24.	30.	149.	28.
75	2	2	440.	2.	3	18.	25.	247.	1.
80	1	25	11063.	33.	1	20.	25.	4149.	830.
80	1	25	11063.	33.	3	20.	25.	6223.	9.
80	2	2	440.	2.	2	25.	30.	149.	25.
80	2	2	440.	2.	3	20.	25.	247.	0.
85	1	25	11063.	33.	1	20.	25.	4149.	830.
85	1	25	11063.	33.	3	20.	25.	6223.	9.
85	2	2	440.	2.	2	25.	30.	149.	25.
85	2	2	440.	2.	3	20.	25.	247.	0.
90	1	25	11063.	33.	1	20.	25.	4149.	830.
90	1	25	11063.	33.	3	20.	25.	6223.	9.
90	2	2	440.	2.	2	25.	30.	149.	25.
90	2	2	440.	2.	3	20.	25.	247.	0.

```
UNITS IN THEATER          25      2      0      0      0      0      0
CARDS PRODUCED      180
FINAL SEQUENCE NUMBERS   255     52      0      0      0      0      0


@PMD.E
PMD NOT ALLOWED


@PRT.I
FURPUR 27R1 RL71-3 08/24/76 19:01:03
UNCLASSIFIED*TPFS(0),F40/30,T
UNCLASSIFIED*32F26(1),F40/30,A   26,
UNCLASSIFIED*32F24(1),F40/30,A   24,
UNCLASSIFIED*32F23(1),F40/30,A   23,
UNCLASSIFIED*32F12(1),F40/30,A   12,
UNCLASSIFIED*58CAMP(1),F14/24,AX   CSINTNAMES,
UNCLASSIFIED*XA(0),F40/30,T   $A,
UNCLASSIFIED*XB(0),F14/24,T   $B,


@ED,R 32F24.
READ-ONLY MODE
CASE UPPER ASSUMED
```

NUR RUN DATE 082476 PAGE 6

```
ED 15.00-08/24/76-19:01:03-(0.)
EDIT
0027A1A     SGPWRS DRY     PKGZ   999000     XPQKXPQKXPQK000000
0027A1W       005000
0028A1A     SAPWRS AMMO    PKGZ   999000     XPQKXPQKXPQK000000
0028A1W       000800
0029A1A     P2PWRS POL     POLZ   999000     XPQKXPQKXPQK000000
0029A1W              000040
0003A2A     SGPWRS DRY     PKGZ   999000     WTGRWTGRWTGR000000
0003A2W       001000
0004A2A     SAPWRS AMMO    PKGZ   999000     WTGRWTGRWTGR000000
0004A2W       000200
0005A2A     P2PWRS POL     POLZ   999000     WTGRWTGRWTGR000000
0005A2W              000001
0031A1A     SARESUPPLY AMMPKGP    010018     AKWQXPQKXPQK000000
0031A1W       000963
0049A1A     SARESUPPLY AMMPKGP    010019     AKWQXPQKXPQK000000
0049A1W       000164
0067A1A     SARESUPPLY AMMPKGP    010020     AKWQXPQKXPQK000000
0067A1W       000583
0006A2A     P2RESUPPLY POLPOLP    010020     HQLHWTGRWTGR000000
0006A2W              000000
0085A1A     SARESUPPLY AMMPKGP    010021     AKWQXPQKXPQK000000
0085A1W       000203
0007A2A     P2RESUPPLY POLPOLP    010021     HQLHWTGRWTGR000000
0007A2W              000000
0103A1A     SARESUPPLY AMMPKGP    010022     AKWQXPQKXPQK000000
0103A1W       000203
0008A2A     P2RESUPPLY POLPOLP    010022     HQLHWTGRWTGR000000
0008A2W              000000
0121A1A     SARESUPPLY AMMPKGP    010023     AKWQXPQKXPQK000000
0121A1W       000658
0009A2A     P2RESUPPLY POLPOLP    010023     HQLHWTGRWTGR000000
0009A2W              000000
0139A1A     SARESUPPLY AMMPKGP    010024     AKWQXPQKXPQK000000
0139A1W       000245
0010A2A     P2RESUPPLY POLPOLP    010024     HQLHWTGRWTGR000000
0010A2W              000000
0157A1A     SARESUPPLY AMMPKGP    010025     AKWQXPQKXPQK000000
0157A1W       000245
0011A2A     P2RESUPPLY POLPOLP    010025     HQLHWTGRWTGR000000
0011A2W              000000
0175A1A     SARESUPPLY AMMPKGP    010026     AKWQXPQKXPQK000000
0175A1W       000701
0012A2A     P2RESUPPLY POLPOLP    010026     HQLHWTGRWTGR000000
0012A2W              000000
0193A1A     SARESUPPLY AMMPKGP    010027     AKWQXPQKXPQK000000
0193A1W       000288
0013A2A     P2RESUPPLY POLPOLP    010027     HQLHWTGRWTGR000000
0013A2W              000000
0210A1A     SGRESUPPLY DRYPKGP    010026     RPCAXPQKXPQK000000
0210A1W       000143
0211A1A     SARESUPPLY AMMPKGP    010026     AKWQXPQKXPQK000000
0211A1W       000824
0212A1A     P2RESUPPLY POLPOLP    010026     HQLHXPQKXPQK000000
0212A1W              000000
0213A1A     SGREPLACEMENTSPKGA    010026     HEKPXPQKXPQK000045
```

VIII-49

```
0014A2A   SARESUPPLY AMMPKGP  010026    AKWQWTGRWTGR000000
0014A2W    000012
0015A2A   P2RESUPPLY POLPOLP  010026    HQLHWTGRWTGR000000
0015A2W              000000
0016A2A   SGREPLACEMENTSPKGA  010026    HEKPWTGRWTGR000004
0214A1A   SGRESUPPLY DRYPKGP  010031    RPCAXPQKXPQK000000
0214A1W    000727
0215A1A   SARESUPPLY AMMPKGP  010031    AKWQXPQKXPQK000000
0215A1W    002061
0216A1A   P2RESUPPLY POLPOLP  010031    HQLHXPQKXPQK000000
0216A1W              000005
0217A1A   SGREPLACEMENTSPKGA  010031    HEKPXPQKXPQK000023
0017A2A   SARESUPPLY AMMPKGP  010031    AKWQWTGRWTGR000000
0017A2W    000027
0018A2A   P2RESUPPLY POLPOLP  010031    HQLHWTGRWTGR000000
0018A2W              000001
0019A2A   SGREPLACEMENTSPKGA  010031    HEKPWTGRWTGR000002
0218A1A   SGRESUPPLY DRYPKGS  010036    RPCAXPQKXPQK000000
0218A1W    000257
0219A1A   P2RESUPPLY POLPOLS  010036    HQLHXPQKXPQK000000
0219A1W              000008
0220A1A   SGREPLACEMENTSPKGA  010036    HEKPXPQKXPQK000022
0020A2A   SARESUPPLY AMMPKGS  010036    AKWQWTGRWTGR000000
0020A2W    000028
0021A2A   P2RESUPPLY POLPOLS  010036    HQLHWTGRWTGR000000
0021A2W              000001
0022A2A   SGREPLACEMENTSPKGA  010036    HEKPWTGRWTGR000002
0221A1A   SGRESUPPLY DRYPKGS  010041    RPCAXPQKXPQK000000
0221A1W    001088
0222A1A   SARESUPPLY AMMPKGS  010041    AKWQXPQKXPQK000000
0222A1W    001282
0223A1A   P2RESUPPLY POLPOLS  010041    HQLHXPQKXPQK000000
0223A1W              000010
0224A1A   SGREPLACEMENTSPKGA  010041    HEKPXPQKXPQK000024
0023A2A   SARESUPPLY AMMPKGS  010041    AKWQWTGRWTGR000000
0023A2W    000028
0024A2A   P2RESUPPLY POLPOLS  010041    HQLHWTGRWTGR000000
0024A2W              000001
0025A2A   SGREPLACEMENTSPKGA  010041    HEKPWTGRWTGR000002
0225A1A   SGRESUPPLY DRYPKGS  010046    RPCAXPQKXPQK000000
0225A1W    000761
0226A1A   SARESUPPLY AMMPKGS  010046    AKWQXPQKXPQK000000
0226A1W    001356
0227A1A   P2RESUPPLY POLPOLS  010046    HQLHXPQKXPQK000000
0227A1W              000006
0228A1A   SGREPLACEMENTSPKGA  010046    HEKPXPQKXPQK000027
0026A2A   SARESUPPLY AMMPKGS  010046    AKWQWTGRWTGR000000
0026A2W    000028
0027A2A   P2RESUPPLY POLPOLS  010046    HQLHWTGRWTGR000000
0027A2W              000001
0028A2A   SGREPLACEMENTSPKGA  010046    HEKPWTGRWTGR000002
0229A1A   SARESUPPLY AMMPKGS  010051    AKWQXPQKXPQK000000
0229A1W    001356
0230A1A   P2RESUPPLY POLPOLS  010051    HQLHXPQKXPQK000000
0230A1W              000006
0231A1A   SGREPLACEMENTSPKGA  010051    HEKPXPQKXPQK000026
0029A2A   SARESUPPLY AMMPKGS  010051    AKWQWTGRWTGR000000
```


NUR RUN   DATE 082476   PAGE   8

```
0029A2W      000028
0030A2A    P2RESUPPLY POLPOLS  010051       HQLHWTGRWTGR000000
0030A2W            000000
0031A2A    SGREPLACEMENTSPKGA  010051       HEKPWTGRWTGR000002
0232A1A    SGRESUPPLY DRYPKGS  010056       RPCAXPQKXPQK000000
0232A1W      001317
0233A1A    SARESUPPLY AMMPKGS  010056       AKWQXPQKXPQK000000
0233A1W      003382
0234A1A    P2RESUPPLY POLPOLS  010056       HQLHXPQKXPQK000000
0234A1W            000020
0235A1A    SGREPLACEMENTSPKGA  010056       HEKPXPQKXPQK000025
0032A2A    SARESUPPLY AMMPKGS  010056       AKWQWTGRWTGR000000
0032A2W      000028
0033A2A    P2RESUPPLY POLPOLS  010056       HQLHWTGRWTGR000000
0033A2W            000001
0034A2A    SGREPLACEMENTSPKGA  010056       HEKPWTGRWTGR000002
0236A1A    SGRESUPPLY DRYPKGS  010061       RPCAXPQKXPQK000000
0236A1W      000823
0237A1A    SARESUPPLY AMMPKGS  010061       AKWQXPQKXPQK000000
0237A1W      001955
0238A1A    P2RESUPPLY POLPOLS  010061       HQLHXPQKXPQK000000
0238A1W            000009
0239A1A    SGREPLACEMENTSPKGA  010061       HEKPXPQKXPQK000033
0035A2A    SARESUPPLY AMMPKGS  010061       AKWQWTGRWTGR000000
0035A2W      000028
0036A2A    P2RESUPPLY POLPOLS  010061       HQLHWTGRWTGR000000
0036A2W            000001
0037A2A    SGREPLACEMENTSPKGA  010061       HEKPWTGRWTGR000002
0240A1A    SGRESUPPLY DRYPKGS  010066       RPCAXPQKXPQK000000
0240A1W      000823
0241A1A    SARESUPPLY AMMPKGS  010066       AKWQXPQKXPQK000000
0241A1W      001955
0242A1A    P2RESUPPLY POLPOLS  010066       HQLHXPQKXPQK000000
0242A1W            000009
0243A1A    SGREPLACEMENTSPKGA  010066       HEKPXPQKXPQK000033
0038A2A    SARESUPPLY AMMPKGS  010066       AKWQWTGRWTGR000000
0038A2W      000028
0039A2A    P2RESUPPLY POLPOLS  010066       HQLHWTGRWTGR000000
0039A2W            000001
0040A2A    SGREPLACEMENTSPKGA  010066       HEKPWTGRWTGR000002
0244A1A    SGRESUPPLY DRYPKGS  010071       RPCAXPQKXPQK000000
0244A1W      000852
0245A1A    P2RESUPPLY POLPOLS  010071       HQLHXPQKXPQK000000
0245A1W            000009
0246A1A    SGREPLACEMENTSPKGA  010071       HEKPXPQKXPQK000033
0041A2A    SARESUPPLY AMMPKGS  010071       AKWQWTGRWTGR000000
0041A2W      000028
0042A2A    P2RESUPPLY POLPOLS  010071       HQLHWTGRWTGR000000
0042A2W            000001
0043A2A    SGREPLACEMENTSPKGA  010071       HEKPWTGRWTGR000002
0247A1A    SGRESUPPLY DRYPKGS  010076       RPCAXPQKXPQK000000
0247A1W      000830
0248A1A    P2RESUPPLY POLPOLS  010076       HQLHXPQKXPQK000000
0248A1W            000009
0249A1A    SGREPLACEMENTSPKGA  010076       HEKPXPQKXPQK000033
0044A2A    SARESUPPLY AMMPKGS  010076       AKWQWTGRWTGR000000
0044A2W      000025
```

```
0045A2A     P2RESUPPLY POLPOLS  010076     HQLHWTGRWTGR000000
0045A2W              000000
0046A2A     SGREPLACEMENTSPKGA  010076     HEKPWTGRWTGR000002
0250A1A     SGRESUPPLY DRYPKGS  010081     RPCAXPQKXPQK000000
0250A1W         000830
0251A1A     P2RESUPPLY POLPOLS  010081     HQLHXPQKXPQK000000
0251A1W              000009
0252A1A     SGREPLACEMENTSPKGA  010081     HEKPXPQKXPQK000033
0047A2A     SARESUPPLY AMMPKGS  010081     AKWQWTGRWTGR000000
0047A2W         000025
0048A2A     P2RESUPPLY POLPOLS  010081     HQLHWTGRWTGR000000
0048A2W              000000
0049A2A     SGREPLACEMENTSPKGA  010081     HEKPWTGRWTGR000002
0253A1A     SGRESUPPLY DRYPKGS  010086     RPCAXPQKXPQK000000
0253A1W         000830
0254A1A     P2RESUPPLY POLPOLS  010086     HQLHXPQKXPQK000000
0254A1W              000009
0255A1A     SGREPLACEMENTSPKGA  010086     HEKPXPQKXPQK000033
0050A2A     SARESUPPLY AMMPKGS  010086     AKWQWTGRWTGR000000
0050A2W         000025
0051A2A     P2RESUPPLY POLPOLS  010086     HQLHWTGRWTGR000000
0051A2W              000000
0052A2A     SGREPLACEMENTSPKGA  010086     HEKPWTGRWTGR000002
EOF:192 SCAN:191
NO CORRECTIONS APPLIED.

@START START.24RTEST.MORSARUN
```

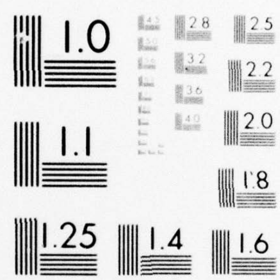
```
@RUN,/PRNT A124E,H3883P5121,UNCLASSIFIED,100,300 . MORSARUN

@HDG   MERGE MORSA CARDS ON ONE TAPE

@QUAL UNCLASSIFIED

@ASG,AX .58CAMP.

@ASG,TIME 8.,8C,SAVEW

@ASG,A  32F10.  . UNIT CARDS

@ASG,A  32F24.  . NON UNIT CARDS

@USE 10.,32F10.

@USE 24.,32F24.

@XQT .58CAMP.MERGFMORSA
 PLANID    2    1    @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
 SERVICE   A    A    A    A    A    A    A    A    A    A    A    A    A    A
 SEQUENCE  0052 0085 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

@MARK 8.
FURPUR 27R1 RL71-3 08/24/76 19:02:40

@START    START.24RTEST.PKGRUN
```

```
@RUN,/PRT A124F,H3883P3043K,UNCLASSIFIED,10,30 . PKGRUN

@HDG   PACKAGE RUN

@QUAL UNCLASSIFIED

@ASG,AX •58CAMP

@ASG,A 32F26.    . MDF2 FILE

@ASG,A 32F23.    . NUR DATA (ORIG,DEST,RDD SEQUENCE)

@ASG,A 32F22.    . SMOBSMOD CARDS - OUTPUT (RDD SEQUENCE)

@ASG,A 32F21.    . PRINT OF UNITS IN PACKAGES

@ASG,A 32F20.    . PRINT OF ONE LINE/PACKAGE

@ASG,A 32F19.    . NUR PACKAGE DATA

@ASG,A  •24TOSV            . OTHER SERVICE PACKAGE DATA

@USE 26.,32F26.

@USE 23.,32F23.

@USE 22.,32F22.

@USE 21.,32F21.
```

AD-A048 554 ARMY CONCEPTS ANALYSIS AGENCY BETHESDA MD F/G 15/7
COMPUTER ASSISTED MATCH PROGRAM (CAMP), (U)
AUG 76 G L MARTIN, E R MONTAGNE
UNCLASSIFIED CAA-D-76-5 NL

4 OF 4
AD
A048554

END
DATE
FILMED
2 -78
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
@USE 20.,32F20.

@USE 19.,32F19.

@USE 18.,*24TOSV.

@PRT.S 24TESTDATA.PKGDATA
FURPUR 27R1 RL71-3 08/24/76 19:03:09
```


PACKAGE RUN DATE 082476 PAGE 3

```
UNCLASSIFIED*24TESTDATA(1).PKGDATA
      1        10  1
      2         1 999 4 999 7 999 8 999 9 999


@XQT *58CAMP.PCKARS


@ADD,P 24TESTDATA,PKGDATA
 DELTA FOR RDD = 10
 DELTA FOR AVL =  1
 THEATER          1      4      7      8      9      0      0
 DAY            999    999    999    999    999      0      0
 UNITS PROCESSED       27
 NUR   PROCESSED      103
 TOT # OF PCKAGES      52
 # OF NUR  PACKS       39
 OTHER SERVICES         0
 MAX UNITS IN PCK       5


@PMD,E


@PRT,I
FURPUR 27R1 RL71-3 08/24/76 19:03:24
UNCLASSIFIED*TPFS(0).F40/30,T
UNCLASSIFIED*58CAMP(1).F14/24,AX   CS;NTNAMES,
UNCLASSIFIED*32F24(1).F40/30,A   26,
UNCLASSIFIED*32F23(1).F40/30,A   23,
UNCLASSIFIED*32F22(1).F40/30,A   22,
UNCLASSIFIED*32F21(1).F40/30,A   21,
UNCLASSIFIED*32F20(1).F40/30,A   20,
UNCLASSIFIED*32F19(1).F40/30,A   19,
UNCLASSIFIED*24TOSY(1).F40/30,A   18,
UNCLASSIFIED*XA(0).F40/30,T   SA,
UNCLASSIFIED*XB(0).F2,T   SB,


@ED,R 32F22.
READ-ONLY MODE
CASE UPPER ASSUMED
ED 15.00-08/24/76-19:03:24-(0,)
EDIT
 PACKAGE 000007              RDD=     2.00       AVL=     .00     1
      .00    4   186    0      .780           1    1
      .00    4   136    0             .387    5    1
      .00    4   186    0             .004    7    1
      .00    4   186    0             .120    8    1
      .00    4   186    0             .365    9    1
      .00    4   186    0             .404   10    1
      .00    4   186    0            1.137   11    1
```

.00	4	186	0			.073	13	1		
.00	4	186	0			.004	14	1		
.00	4	186	0			.214	15	1		
5444.				0.		0.				
PACKAGE 000012					RDD=	2.00		AVL=	.00	2
.00	4	227	0	.780			1	2		
.00	4	227	0			.387	5	2		
.00	4	227	0			.004	7	2		
.00	4	227	0			.120	8	2		
.00	4	227	0			.365	9	2		
.00	4	227	0			.404	10	2		
.00	4	227	0			1.137	11	2		
.00	4	227	0			.073	13	2		
.00	4	227	0			.004	14	2		
.00	4	227	0			.214	15	2		
5444.				0.		0.				
PACKAGE 000003					RDD=	5.00		AVL=	.00	3
.00	4	178	0	.863			1	3		
.00	4	178	0			1.760	5	3		
.00	4	178	0			.011	7	3		
.00	4	178	0			.636	10	3		
.00	4	178	0			.820	11	3		
.00	4	178	0			.004	12	3		
.00	4	178	0			.232	13	3		
.00	4	178	0			.006	14	3		
.00	4	178	0			.245	15	3		
5444.				0.		0.				
PACKAGE 000008					RDD=	5.00		AVL=	.00	4
.00	4	210	0	.851			1	4		
.00	4	210	0			1.760	5	4		
.00	4	210	0			.011	7	4		
.00	4	210	0			.617	10	4		
.00	4	210	0			.820	11	4		
.00	4	210	0			.004	12	4		
.00	4	210	0			.232	13	4		
.00	4	210	0			.006	14	4		
.00	4	210	0			.243	15	4		
5444.				0.		0.				
PACKAGE 000013					RDD=	10.00		AVL=	10.00	5
10.00	12	190	0	.440			1	5		
10.00	12	190	0			1.068	4	5		
10.00	12	190	0			.025	5	5		
10.00	12	190	0			.480	8	5		
10.00	12	190	0			2.920	9	5		
10.00	12	190	0			.295	10	5		
10.00	12	190	0			.076	11	5		
10.00	12	190	0			.001	12	5		
10.00	12	190	0			.032	13	5		
10.00	12	190	0			.001	14	5		
10.00	12	190	0			.115	15	5		
5444.				0.		0.				
PACKAGE 000004					RDD=	17.00		AVL=	.00	6
.00	4	178	0	.006			1	6		
.00	4	178	0			.009	10	6		
.00	4	178	0			.001	15	6		
5444.				0.		0.				
PACKAGE 000014					RDD=	18.00		AVL=	10.00	7

```
  10.00    4  174   0               5.077   17   -7
 5444.                   0.         0.
PACKAGE 000009                RDD=   20.00       AVL=     .00     8
    .00    4  210   0    2.474              1    8
    .00    4  210   0               3.908   5    8
    .00    4  210   0                .027   7    8
    .00    4  210   0                .120   8    8
    .00    4  210   0                .365   9    8
    .00    4  210   0               1.630  10    8
    .00    4  210   0               2.777  11    8
    .00    4  210   0                .008  12    8
    .00    4  210   0                .536  13    8
    .00    4  210   0                .016  14    8
    .00    4  210   0                .699  15    8
 5444.                   0.         0.
PACKAGE 000041                RDD=   20.00       AVL=   10.00     9
 5444.                   0.         0.
PACKAGE 000025                RDD=   26.00       AVL=   10.00    10
  10.00    4  187   1     .090              1   -10
 5444.                   0.         0.
PACKAGE 000035                RDD=   26.00       AVL=   10.00    11
  10.00    4  191   0               5.000   6   -11
 5444.                   0.         0.
PACKAGE 000036                RDD=   26.00       AVL=   10.00    12
  10.00    4  191   2               8.000   6   -12
 5444.                   0.         0.
PACKAGE 000047                RDD=   26.00       AVL=   10.00    13
  10.00    4  216   0                .870  16   -13
 5444.                   0.         0.
PACKAGE 000048                RDD=   26.00       AVL=   10.00    14
  10.00    4  216   2                .257  16   -14
 5444.                   0.         0.
PACKAGE 000019                RDD=   26.00       AVL=   10.00    15
  10.00   12  174   0                .039  17   -15
 5444.                   0.         0.
PACKAGE 000020                RDD=   26.00       AVL=   10.00    16
  10.00   12  174   2                .028  17   -16
 5444.                   0.         0.
PACKAGE 000030                RDD=   26.00       AVL=   10.00    17
  10.00   12  187   1     .008              1   -17
 5444.                   0.         0.
PACKAGE 000005                RDD=   29.00       AVL=     .00    18
    .00    4  178   0     .018              1    18
    .00    4  178   0                .028  10    18
    .00    4  178   0                .003  15    18
 5444.                   0.         0.
PACKAGE 000015                RDD=   31.00       AVL=   10.00    19
  10.00    4  174   0               2.061  17   -19
 5444.                   0.         0.
PACKAGE 000016                RDD=   31.00       AVL=   10.00    20
  10.00    4  174   2               1.282  17   -20
 5444.                   0.         0.
PACKAGE 000042                RDD=   31.00       AVL=   10.00    21
  10.00   12  191   0               1.000   6   -21
 5444.                   0.         0.
PACKAGE 000043                RDD=   31.00       AVL=   10.00    22
  10.00   12  191   2               2.000   6   -22
```

PACKAGE RUN DATE 082476 PAGE 6

```
 5444.                          0.        0.
PACKAGE 000010                          RDD=   38.00          AVL=     .00     23
      .00     4   210    0    .993                    1   23
      .00     4   210    0              1.962         5   23
      .00     4   210    0               .011         7   23
      .00     4   210    0               .365         9   23
      .00     4   210    0               .741        10   23
      .00     4   210    0               .856        11   23
      .00     4   210    0               .004        12   23
      .00     4   210    0               .236        13   23
      .00     4   210    0               .007        14   23
      .00     4   210    0               .303        15   23
 5444.                          0.        0.
PACKAGE 000026                          RDD=   41.00          AVL=   10.00     24
    10.00     4   187    1    .077                    1  -24
 5444.                          0.        0.
PACKAGE 000037                          RDD=   41.00          AVL=   10.00     25
    10.00     4   191    2             22.000         6  -25
 5444.                          0.        0.
PACKAGE 000049                          RDD=   41.00          AVL=   10.00     26
    10.00     4   216    2              1.849        16  -26
 5444.                          0.        0.
PACKAGE 000021                          RDD=   41.00          AVL=   10.00     27
    10.00    12   174    2               .084        17  -27
 5444.                          0.        0.
PACKAGE 000031                          RDD=   41.00          AVL=   10.00     28
    10.00    12   187    1    .006                    1  -28
 5444.                          0.        0.
PACKAGE 000006                          RDD=   44.00          AVL=     .00     29
      .00     4   179    0    .847                    1   29
      .00     4   178    0              1.760         5   29
      .00     4   178    0               .011         7   29
      .00     4   178    0               .613        10   29
      .00     4   178    0               .820        11   29
      .00     4   178    0               .004        12   29
      .00     4   178    0               .232        13   29
      .00     4   178    0               .006        14   29
      .00     4   178    0               .242        15   29
 5444.                          0.        0.
PACKAGE 000017                          RDD=   46.00          AVL=   10.00     30
    10.00     4   174    2              6.094        17  -30
 5444.                          0.        0.
PACKAGE 000044                          RDD=   46.00          AVL=   10.00     31
    10.00    12   191    2              2.000         6  -31
 5444.                          0.        0.
PACKAGE 000027                          RDD=   56.00          AVL=   10.00     32
    10.00     4   187    1    .091                    1  -32
 5444.                          0.        0.
PACKAGE 000038                          RDD=   56.00          AVL=   10.00     33
    10.00     4   191    2             38.000         6  -33
 5444.                          0.        0.
PACKAGE 000050                          RDD=   56.00          AVL=   10.00     34
    10.00     4   216    2              2.963        16  -34
 5444.                          0.        0.
PACKAGE 000022                          RDD=   56.00          AVL=   10.00     35
    10.00    12   174    2               .084        17  -35
 5444.                          0.        0.
```

```
PACKAGE 000032                      RDD=    56.00         AVL=    10.00    36
    10.00   12   187    1      .006                     1  -36
  5444.                       0.         0.
PACKAGE 000001        18003         RDD=    59.00         AVL=      .00    37
       .00    4  210    1     3.253                     1   37
       .00    4  210    1                4.100          5   37
       .00    4  210    1                2.400          8   37
       .00    4  210    1                6.205          9   37
       .00    4  210    1                2.287         10   37
       .00    4  210    1                3.439         11   37
       .00    4  210    1                 .195         13   37
       .00    4  210    1                 .016         14   37
       .00    4  210    1                 .724         15   37
  5444.                       0.         0.
PACKAGE 000011                      RDD=    59.00         AVL=      .00    38
       .00    4  210    0      .107                     1   38
       .00    4  210    0                 .020          5   38
       .00    4  210    0                 .032         10   38
       .00    4  210    0                 .553         11   38
       .00    4  210    0                 .025         15   38
  5444.                       0.         0.
PACKAGE 000018                      RDD=    61.00         AVL=    10.00    39
    10.00    4   174    2                3.910         17  -39
  5444.                       0.         0.
PACKAGE 000045                      RDD=    61.00         AVL=    10.00    40
    10.00   12   191    2                3.000          6  -40
  5444.                       0.         0.
PACKAGE 000028                      RDD=    71.00         AVL=    10.00    41
    10.00    4   187    1      .099                     1  -41
  5444.                       0.         0.
PACKAGE 000039                      RDD=    71.00         AVL=    10.00    42
    10.00    4   191    2               27.000          6  -42
  5444.                       0.         0.
PACKAGE 000051                      RDD=    71.00         AVL=    10.00    43
    10.00    4   216    2                2.512         16  -43
  5444.                       0.         0.
PACKAGE 000023                      RDD=    71.00         AVL=    10.00    44
    10.00   12   174    2                 .078         17  -44
  5444.                       0.         0.
PACKAGE 000033                      RDD=    71.00         AVL=    10.00    45
    10.00   12   187    1      .006                     1  -45
  5444.                       0.         0.
PACKAGE 000002                      RDD=    75.00         AVL=      .00    46
       .00    4  192    1      .091                     1   46
       .00    4  192    1                 .718          8   46
       .00    4  192    1                1.460          9   46
       .00    4  192    1                 .002         13   46
       .00    4  192    1                 .004         15   46
  5444.                       0.         0.
PACKAGE 000046                      RDD=    76.00         AVL=    10.00    47
  5444.                       0.         0.
PACKAGE 000029                      RDD=    86.00         AVL=    10.00    48
    10.00    4   187    1      .033                     1  -48
  5444.                       0.         0.
PACKAGE 000040                      RDD=    86.00         AVL=    10.00    49
    10.00    4   191    2                9.000          6  -49
  5444.                       0.         0.
```

```
PACKAGE 000052                          RDD=   86.00      AVL=   10.00    50
   10.00    4  216    2                      .830   16  -50
  5444.                      0.         0.
PACKAGE 000024                          RDD=   86.00      AVL=   10.00    51
   10.00   12  174    2                      .025   17  -51
  5444.                      0.         0.
PACKAGE 000034                          RDD=   86.00      AVL=   10.00    52
   10.00   12  187    1          .002                1  -52
  5444.                      0.         0.
EOF:245 SCAN:244
NO CORRECTIONS APPLIED.

@ED,R 32F20.
READ-ONLY MODE
CASE UPPER ASSUMED
ED 15.00-08/24/76-19:03:26-(0,)
EDIT
```

PACK	DEST	ORIG	MODE	RDD	AVL	# UNIT	TOT PAX	TOT STON	SQ FT/BRL	
1	4	210	1	59.	0.	5	3.253	10.761	8.605	18003
2	4	192	1	75.	0.	1	.091	.006	2.178	******
3	4	178	0	5.	0.	4	.863	3.715	.000	******
4	4	178	0	17.	0.	1	.006	.010	.000	******
5	4	178	0	29.	0.	3	.018	.031	.000	******
6	4	178	0	44.	0.	1	.847	3.689	.000	******
7	4	186	0	2.	0.	1	.780	2.223	.485	******
8	4	210	0	5.	0.	2	.851	3.694	.000	******
9	4	210	0	20.	0.	3	2.474	9.601	.485	******
10	4	210	0	38.	0.	2	.993	4.120	.365	******
11	4	210	0	59.	0.	1	.107	.630	.000	******
12	4	227	0	2.	0.	1	.780	2.223	.485	******
13	12	190	0	10.	10.	2	.440	.546	4.468	******
14	4	174	0	18.	10.	11	.000	5.077	.000	-1
15	4	174	0	31.	10.	1	.000	2.061	.000	-2
16	4	174	2	31.	10.	1	.000	1.282	.000	-3
17	4	174	2	46.	10.	3	.000	6.094	.000	-4
18	4	174	2	61.	10.	2	.000	3.910	.000	-5
19	12	174	0	26.	10.	2	.000	.039	.000	-6
20	12	174	2	26.	10.	1	.000	.028	.000	-7
21	12	174	2	41.	10.	3	.000	.084	.000	-8
22	12	174	2	56.	10.	3	.000	.084	.000	-9
23	12	174	2	71.	10.	3	.000	.078	.000	-10
24	12	174	2	86.	10.	1	.000	.025	.000	-11
25	4	187	1	26.	10.	3	.090	.000	.000	-12
26	4	187	1	41.	10.	3	.077	.000	.000	-13
27	4	187	1	56.	10.	3	.091	.000	.000	-14
28	4	187	1	71.	10.	3	.099	.000	.000	-15
29	4	187	1	86.	10.	1	.033	.000	.000	-16
30	12	187	1	26.	10.	3	.008	.000	.000	-17
31	12	187	1	41.	10.	3	.006	.000	.000	-18
32	12	187	1	56.	10.	3	.006	.000	.000	-19
33	12	187	1	71.	10.	3	.006	.000	.000	-20
34	12	187	1	86.	10.	1	.002	.000	.000	-21
35	4	191	0	26.	10.	2	.000	.000	5.000	-22
36	4	191	2	26.	10.	1	.000	.000	8.000	-23
37	4	191	2	41.	10.	3	.000	.000	22.000	-24

38	4	191	2	56.	10.	3	.000	.000	38.000	-25
39	4	191	2	71.	10.	3	.000	.000	27.000	-26
40	4	191	2	86.	10.	1	.000	.000	9.000	-27
41	12	191	0	20.	10.	9	.000	.000	.000	-28
42	12	191	0	31.	10.	1	.000	.000	1.000	-29
43	12	191	2	31.	10.	2	.000	.000	2.000	-30
44	12	191	2	46.	10.	3	.000	.000	2.000	-31
45	12	191	2	61.	10.	3	.000	.000	3.000	-32
46	12	191	2	76.	10.	3	.000	.000	.000	-33
47	4	216	0	26.	10.	2	.000	.870	.000	-34
48	4	216	2	26.	10.	1	.000	.257	.000	-35
49	4	216	2	41.	10.	2	.000	1.849	.000	-36
50	4	216	2	56.	10.	3	.000	2.963	.000	-37
51	4	216	2	71.	10.	3	.000	2.512	.000	-38
52	4	216	2	86.	10.	1	.000	.830	.000	-39

EOF:53 SCAN:52
NO CORRECTIONS APPLIED.

@HDG PRINT OF UNIT PACKAGES

@XQT .58CAMP.PRTPCK

PRINT OF UNIT PACKAGES DATE 082476 PAGE 11

PACKAGE DEST ORIG MODE RDD AVL UNITS TOT PAX TOT STON TOT SQ FT (1000'S OF PAX, TONS, SQ FT)

PACKAGE	DEST	ORIG	MODE	RDD	AVL	UNITS	TOT PAX	TOT STON	TOT SQ FT
1	4	210	1	59	0	5	3.253	10.761	8.605
2	4	192	1	75	0	1	.091	.006	2.178
3	4	178	0	5	0	4	.863	3.715	.000
4	4	178	0	17	0	1	.006	.010	.000
5	4	178	0	29	0	3	.018	.031	.000
6	4	178	0	44	0	1	.847	3.689	.000
7	4	186	0	2	0	1	.780	2.223	.485
8	4	210	0	5	0	2	.851	3.694	.000
9	4	210	0	20	0	3	2.474	9.601	.485
10	4	210	0	38	0	2	.993	4.120	.365
11	4	210	0	59	0	1	.107	.630	.000
12	4	227	0	2	0	1	.780	2.223	.485
13	12	190	0	10	10	2	.440	.546	4.468

PACKAGE	SRC	UIC	TPSN	SRC	UIC	TPSN	SRC	UIC	TPSN	SRC	UIC	TPSN
1	17387H50000	TEST09	18003	17055H01000	TEST07	18003	17055H01000	TEST06	18003	17055H01000	TEST08	18003
1	17052H00000	TEST05	18003									
2	05079H40000	TEST20	21435									
3	05115H30000	TEST15	21364	05510H2FA01	X50002	31572	05510H2FB01	X50003	31300	05510H2FB01	X50004	31300
4	05510H2FB01	X50005	31300									
5	05510H2FB01	X50006	31300	05510H2FB01	X50007	31300	05510H2FB01	X50008	31300			
6	05115H50000	X50001	31300									
7	05035H50000	TEST10	21353									
8	05115H30000	TEST16	21364	05510H2FA00	TEST23	31557						
9	05035H50000	TEST13	21353	05115H30000	TEST17	21364	05115H30000	TEST18	21364			
10	05115H30000	TEST19	21364	05327G60008	TEST21	31531						
11	05124G60000	TEST24	31436									
12	05035H50000	TEST11	21353									
13	01127H10000	TEST03	30623	01207H20000	TEST04	30612						

PRINT OF UNIT PACKAGES DATE 082476 PAGE 11

PACKAGE	DEST	ORIG	MODE	RDD	AVL	ENTRY	TOT PAX	TOT STON	TOT BARRL (1000'S OF PAX, TONS, BARRELS
14	4	174	0	18	10	11	.000	5.077	.000
15	4	174	0	31	10	1	.000	2.061	.000
16	4	174	2	31	10	1	.000	1.282	.000
17	4	174	2	46	10	3	.000	6.094	.000
18	4	174	2	61	10	2	.000	3.910	.000
19	12	174	0	26	10	2	.000	.039	.000
20	12	174	2	26	10	1	.000	.028	.000
21	12	174	2	41	10	3	.000	.084	.000
22	12	174	2	56	10	3	.000	.084	.000
23	12	174	2	71	10	3	.000	.078	.000
24	12	174	2	86	10	1	.000	.025	.000
25	4	187	1	26	10	3	.090	.000	.000
26	4	187	1	41	10	3	.077	.000	.000
27	4	187	1	56	10	3	.091	.000	.000
28	4	187	1	71	10	3	.099	.000	.000
29	4	187	1	86	10	1	.033	.000	.000
30	12	187	1	26	10	3	.008	.000	.000
31	12	187	1	41	10	3	.006	.000	.000
32	12	187	1	56	10	3	.006	.000	.000
33	12	187	1	71	10	3	.006	.000	.000
34	12	187	1	86	10	1	.002	.000	.000
35	4	191	0	26	10	2	.000	.000	5.000
36	4	191	2	26	10	1	.000	.000	8.000
37	4	191	2	41	10	3	.000	.000	22.000
38	4	191	2	56	10	3	.000	.000	38.000
39	4	191	2	71	10	3	.000	.000	27.000

PRINT OF UNIT PACKAGES DATE 082476 PAGE 12

PACKAGE	DEST	ORIG	MODE	RDD	AVL	ENTRY	TOT PAX	TOT STON	TOT BARRL (1000'S OF PAX, TONS, BARRELS
40	4	191	2	86	10	1	.000	.000	9.000
41	12	191	0	20	10	9	.000	.000	.000
42	12	191	0	31	10	1	.000	.000	1.000
43	12	191	2	31	10	2	.000	.000	2.000
44	12	191	2	46	10	3	.000	.000	2.000
45	12	191	2	61	10	3	.000	.000	3.000
46	12	191	2	76	10	3	.000	.000	.000
47	4	216	0	26	10	2	.000	.870	.000
48	4	216	2	26	10	1	.000	.257	.000
49	4	216	2	41	10	2	.000	1.849	.000
50	4	216	2	56	10	3	.000	2.963	.000
51	4	216	2	71	10	3	.000	2.512	.000
52	4	216	2	86	10	1	.000	.830	.000

@MSG.N START•24R.LAYINRUN

PRINT OF UNIT PACKAGES DATE 082476 PAGE 13

78